# WORK-BREAKDOWN STRUCTURE

A
Simple and Powerful Tool
for
Project Management

# THE TERMS:
# WORK, BREAKDOWN, AND STRUCTURE

## Work vs. Project:

- Project is work with a very concrete and focussed *goal*. WBS applies to any work/project, small or big.

## Goal vs. Objective

- Goals are what you expect/plan to do and objectives are what you expect to happen as consequences of what you do.
  - The objective "create a shaded area in the yard" can be achieved by doing several very different things (goals):
    (a) putting a tree that will grow big over time, or
    (b) putting up a man-made structure (a roofed area, often called a "shade" like a tin-shade or a fiberglass shade).
- Goals $\leftrightarrow$ Objectives is a many-to-many relation.
- Objectives are stated typially in a detailed and itemized form to elaborate/explain a short and broadly stated goal.
  - Large shaded area vs. 40×100 sq ft. shaded area for ···.

## Breakdown:

- Any large project needs to be broken down into smaller pieces.
  - The project is completed piece by piece.
  - The breakdown helps to clarify the project-work as a whole, plan and control the work-progress, assign manpower and other resources, and review progress.

## Structure:

- Whenever we decompose a larger task into subtasks, the structure arises naturally out of logical relationship among the subtasks.

# AN ABSTRACT VIEW OF ALGORITHMS

**Algorithm at A Highest Abstraction Level:**

- A sequence of atomic actions.

**Theorem.**

- Every thing that happens over a period of time is an algorithm.

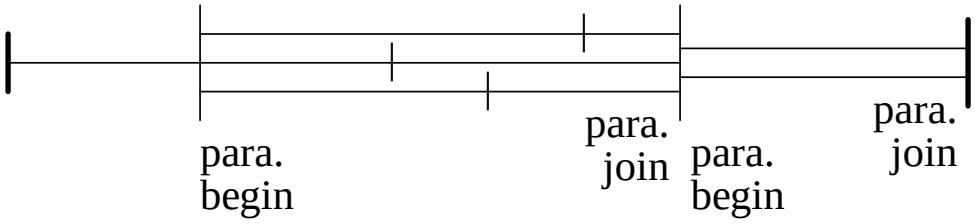
**Finding an Algorithm:**

(1) Finding the atomic actions.

(2) Arranging them in a sequence in some order.

**Where Does WBS Fit In:** Finding the actions.

**What is A Parallel Algorithm:**




**Does WBS apply to Parallel Algorithm:** Yes, Just as much.

**Role of WBS in Software Engineering:**

- Software Engineering can be viewed as making a product combining new and known algorithms.
- The role of WBS: design of high level *algorithms for developing* large scale complex software products.

# WBS IS NOT PRODUCT-REQUIREMENTS

**Requirements:**

- Properties of the product from the *customer's viewpoint*, i.e., what it does for the customer or how it meets the customer's needs.

**WBS:** Comes after the requirements are defined.

- Identifies all activities (work-units) that the product-developer will carry out to deliver the final product.
- Helps to create the project-plan.
  - *Activity definitions:* What things are to be done.
  - *Activity sequencing:* When to do what (logical precedences).
  - *Activity resource allocation (cost/time estimation):* Who will do what and when.
  - *Activity scheduling:* Actual time-line for activities.
  - *Schedule control:* Who will be responsible for what.

**Activity vs Requirement:**

- What actions can be taken to meet the requirement "there should be more space in a classroom per student"?

**Some Useful Sayings:**

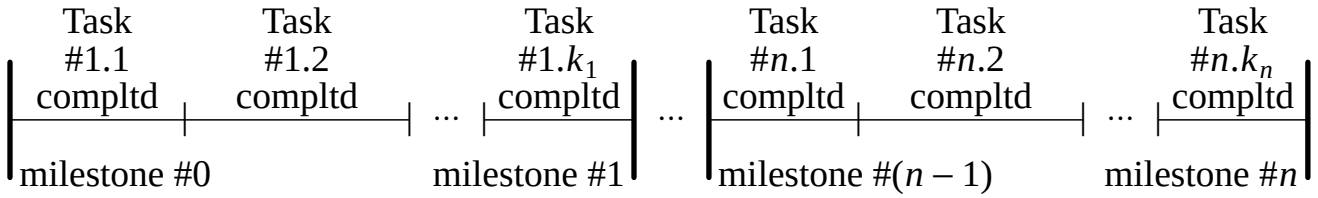
- If you fail to plan then you (essentially) plan to fail.
- The 6P-rule: Prior Planning Prevents Poor Project Performance.

**Question:**

•? What are some ways that a project's performance can be poor?

# AN ABSTRACT VIEW OF THE TIME-LINE OF A PROJECT

**A Simple View:** No overlapping tasks.




**Milestones:**

- Time points for project reviews/evaluations/error-checking before going into the phase.
  - identify bottlenecks and management problems.
  - update the remainder of schedule and resource allocation.
- Error or change-propagation beyond a milestone are avoided.

**Question:**

•? How to determine a time-line (project plan = tasks + schedule)?

- Identify the tasks, their precedence constraints.
- Allocate time-duration (and manpower, equipments, etc).
- Schedule (order) the tasks to satisfy precedence constraints.

•? How to compare different schedules and select the best?

**We Are Not Looking At:**

- The management decision-process for approving a project.

**Question:** What would be a further abstraction of the above view? How would a more complex (general) view differ from the simple view?

# A SCHEDULING PROBLEM AND ITS WBS

**Constraints:**

- One person to repair the cars $C_i$, where $r_i$ = repairTime($C_i$), $1 \le i \le n$. Assume that the cars can be repaired in any order.

**Objective:**

- Minimize total lost service-time (LST) of the cars as they wait for repair.

**Question:**

•? Give all possible repair-schedules and the best repair-schedule for $n = 3$ and $r(C_1) = 2$, $r(C_2) = 7$, and $r(C_3) = 3$.

•? Give a formula to compute LST for the schedule $\langle C_1, C_2, \cdots, C_n \rangle$.

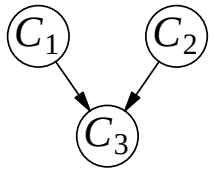
•? Give a general method (algorithm) to find an optimum schedule.

**A Variation** (With Precedence Constraints):

- *Precedence* constraint: which car can be repaired before which.

**Question:**

•? Give the optimal schedule for the precedence constraints:

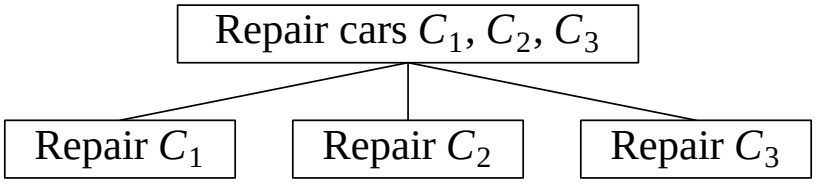



$r(C_1) = 2$
$r(C_2) = 7$
$r(C_3) = 3$

•? What makes this variation more general than the original form?

•? Give an algorithm for the optimum schedule for this variation.

# WBS FOR THE CAR-REPAIR PROBLEM



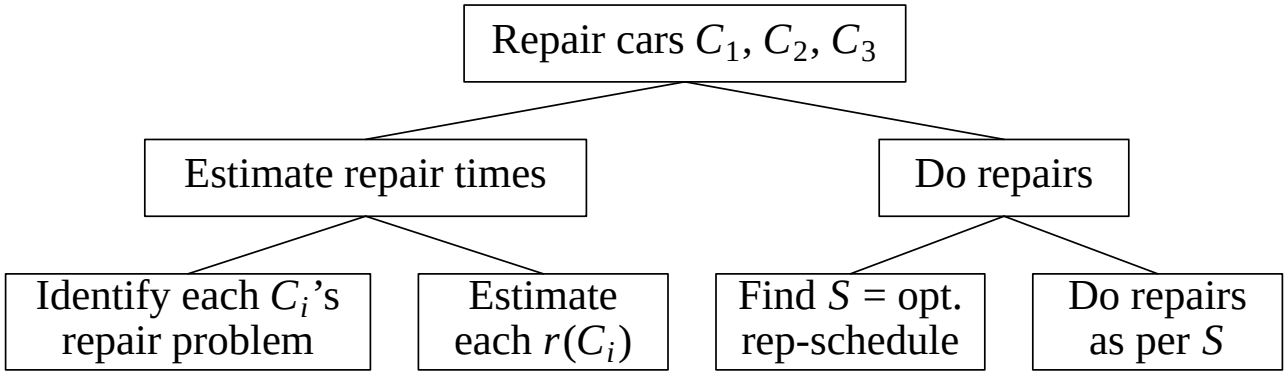

**Role Of Breakdown:**

- Suggests the *possibility* that the cars can perhaps be repaired one at a time instead of multiple cars being repaired in parallel.
- It does not say the order in which the cars are to be repaired.

**Question:** What could be a further breakdown of each "Repair $C_i$"?

**Another WBS:**

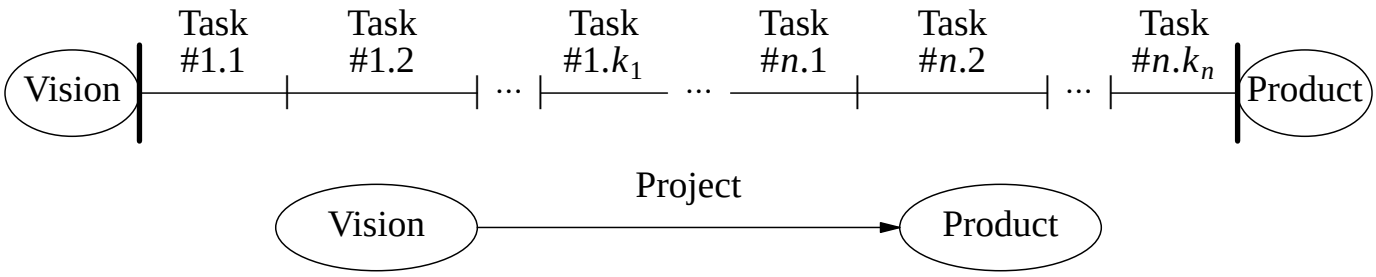
- It has some new work-units and a different overall organization.
- The repair-schedule is not a schedule of work-units in the WBS. (A schedule of work-units in a WBS is not considered part of it because precedence constraints of work-units is not part of WBS.)




**Question:**

•? Show a breakdown of "Identify ···" and "Estimate ···" and the AON of the new WBS; reorganize the new WBS based on the AON.

# PROJECT: THE BRIDGE FROM VISION TO PRODUCT




**Vision:**

- The "idea", the "spark" that gave birth to the project.
- Every project, large and small, has such a vision.

**Question:**

•? What is a key distinction between a leader and a manager?


•? What is a key distinction between a manager and an engineer?

•? How does the programming challenge for a beginning programmer differ from that of an experienced software engineer?

**Multiple Roles:**

- Each of us play each of the roles of leader, manager, and engineer at different times in different degrees.

# SUCCESSFUL SOFTWARE PROJECT AND THE ROLE OF WBS

## Two Attributes of Successful Projects:

(1) A successful product is developed (irrespective of the means used to get to the final product).

(2) A successful development of a product.

**Question:** What is the difference, and what are their significance?

> **What We Really Want:** (1)+(2)
> A successful development of a successful product.

## Some Common Attributes of Successful People:

- Have a clear goal; a clear idea of what they want.
- Pursue them diligently, using all available resources (time, effort, money, knowledge, advice/criticisms) optimally.

## WBS: Work Breakdown Structure

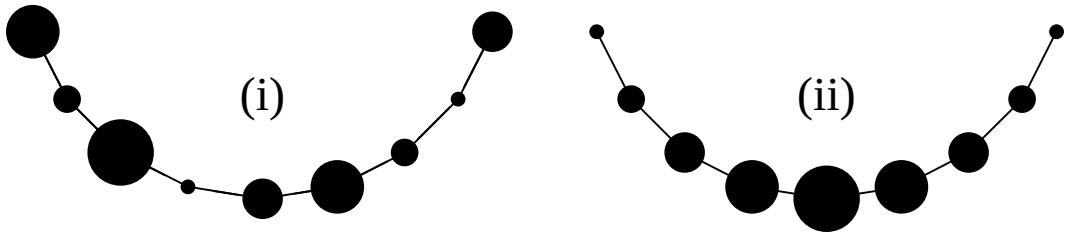
- A technique for successful development of a project-schedule/plan, a prerequisite of part (2).

## EXERCISE

1. State a good product/project idea, with an explanation/justification of why you think the product would be successful. (Indicate novelty of the product/service, need for it, and the likely users/buyers, and compare related competitive products (if any).)
2. Is there a WBS for the task "come up with a good product idea"?
3. Is it true that successful people focus more on how to achieve the goals instead of why it cannot be achieved?

# TWO BASIC RULES
# IN MAKING THINGS BETTER

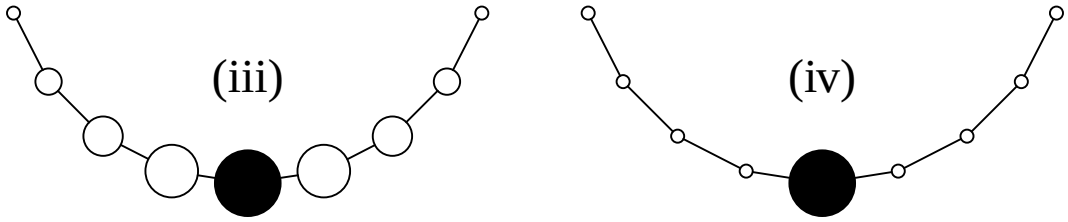
**Which Necklace is More Attractive:**




**Rule #1:** *Reorganize* what you have.

**Question:** What can make applying this rule diffi cult?

**Which of These Two:**




**Rule #2:** *Innovate*; add new pieces or eliminate some.

**Question:**

- •? What is a common property (an abstraction) of all four necklaces?
- •? To change a program $P$ which sorts an input array in increasing order to a program $P'$ which sorts an input array in decreasing order, which rule would be applicable; explain exactly how.
- •? Give an example of Rule#2 application in improving a software.

# EXERCISE

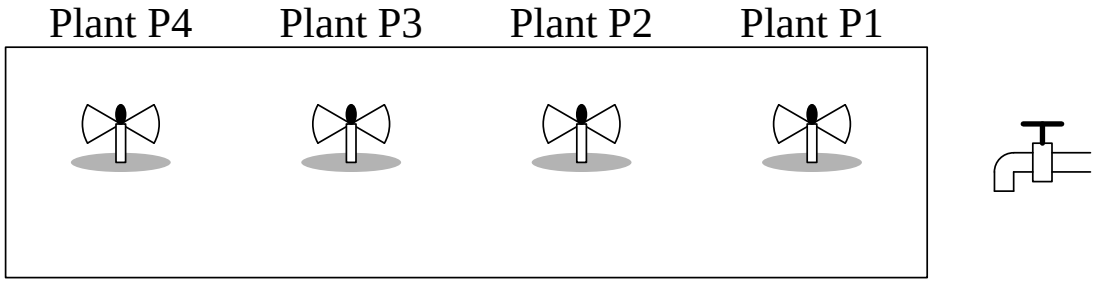
1. Assume that we want to water the four plants $P_j$, $1 \leq j \leq 4$, as shown below in one run using a bucket of capacity 4 cups. Assume that:

   (1) Each plant $P_j$ is to be given 1 cup of water,

   (2) The distance of $P_j$ from the tap on the right is $x_j$, where $x_j > x_{j-1}$, and

   (3) The bucket has weight 0.

   There are many different orders in which we can water the plants, and they require a different amount of effort in carrying the water. What is the best and the worst possible orders to water the plants?




2. In what way, the reorganization (reordering) rule applies here?
3. How does the optimal solution change if we were to make multiple runs, and, in particular, we are allowed carry just one cup of water at a time?
4. Is there any advantage in using the bucket and one run to water all the plants instead of making multiple runs?
5. Give another application of reorganization/reordering rule.
6. What could be an innovation in the context of Problem 1?

# WBS:
# WORK-BREAKDOWN-STRUCTURE

Most things are composition two or more simpler things.

**What is it?**

- An exhaustive and objective list of things-to-do, in a simple case.
- For a complex project/task, it is a hierarchical (tree) structure, whose terminal nodes show all *atomic* work-units to meet the project deliverables.
- Gives a clear and complete picture of the *scope* of work.

**Why do we need it?**

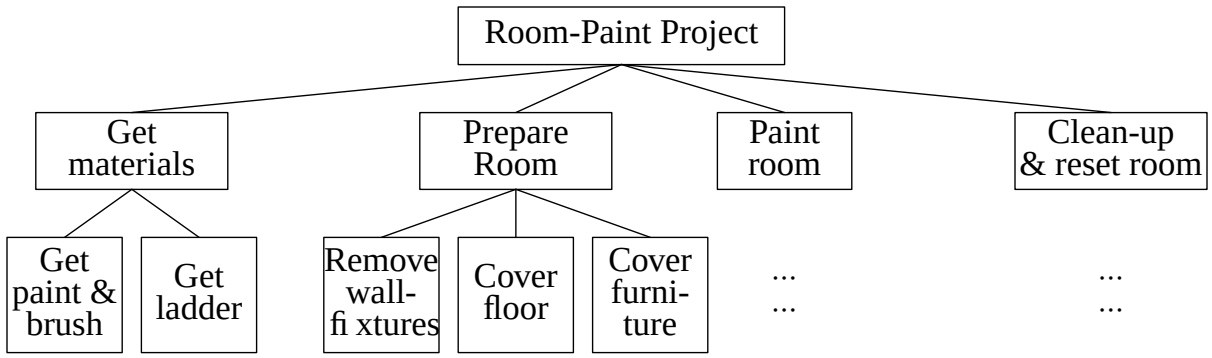
- Understanding the work-units comprising the total project work is a minimum requirement for successful completion of a project.
- Needed for developing the Activity Network model, which is an important project planning and management tool:
  - allocate resources (time + personnel) and create a schedule.
  - monitor work-progress to meet milestones and the project deadline, and taking corrective actions as needed.

**How to Create it?** To be described here.

**Question:** Successful people also do small things, but there is a difference between the small things done by a successful person vs. the small things done by others? What is it?

# AN EXAMPLE WBS

**Paint-Room-WBS** (a template):




**Work-Unit:** A terminal node of the tree-hierarchy.

- One person responsible for each work-unit's proper completion.

**Top-Down Approach:**

- Best way to identify the work-units is by the top-down and breadth-fi rst (level by level).

**Question:**

•? What would be a possible breakdown of "Paint room" and what are its advantages?

•? Can we expand "Get materials" more? If we merge it with its children, how to rename it (without loss of information)?

•? If we rename "Get paint & brush" as "Buy paint & brush" does it make any difference? Will it be worthwhile to break it further, and what will a breakdown look like.

•? How will the template Paint-Room-WBS differ from the WBS for painting a specifi c room?

•? Show a WBS for creating a home flower-garden.

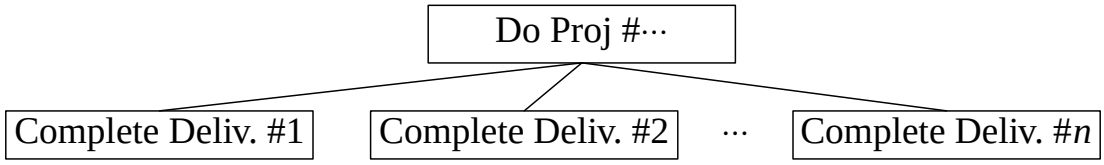
# A POOR WBS

**Work-units vs. Deliverables:**

- They are not the same.
- Project deliverables are part of project requirements, and work-units are part of WBS.

**Example of Poor WBS:**

- Does not identify the work-units to meet the deliverables.




**Typical Work-Unit Types in a Completed Software:**

- Creating/reading input, processing input, or creating output.

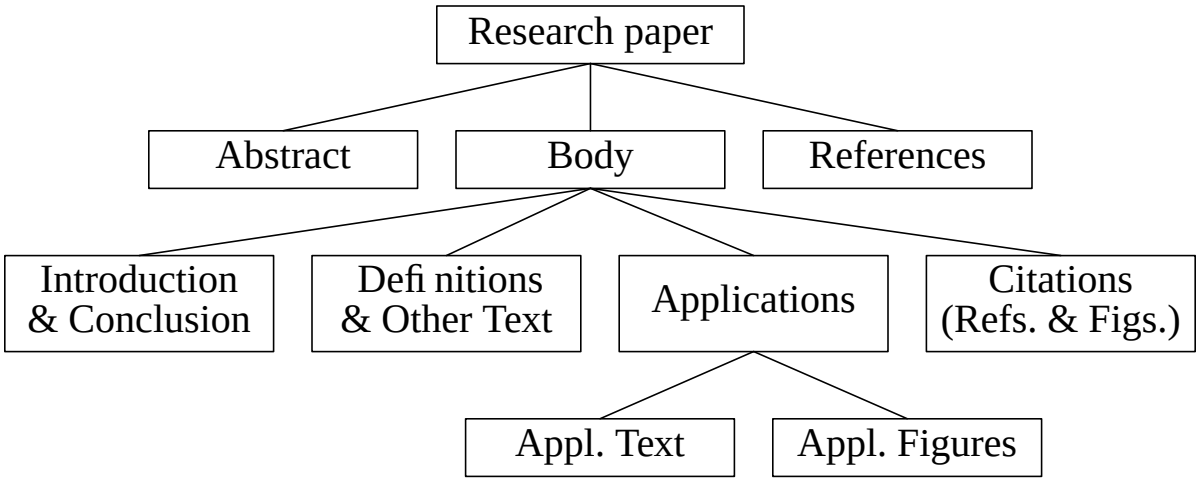
**Question:**

•? Name some common work-units in developing softwares.

•? What are some work-units in an operating system and a compiler?

# WHAT'S IN A WBS TREE

**Example of WBS** (for writing a particular research paper):




(The part "Definitions & Other Texts" can also be broken down further, and have Examples and Figures related to definitions.)

**Features of WBS-tree** (Completeness = 100% rule):

- Higher level nodes tend to be "work-types" or "work-categories" (hence common to many WBS's).
- The terminal nodes, also called *work-units*, are more project specific. They together represent the actual total work to be done (100% rule), and account for all internal and external deliverables, and all input (material/data) uses and outputs created.
- Each intermediate node is just a larger conceptual "chunk of work", the children (≥ 2) give a breakdown to smaller chunks.
- A work-unit is a small enough chunk that is measurable (can estimate resources - time, personnel, costs - needed to complete it).
- The intermediate nodes help to identify the terminal nodes.

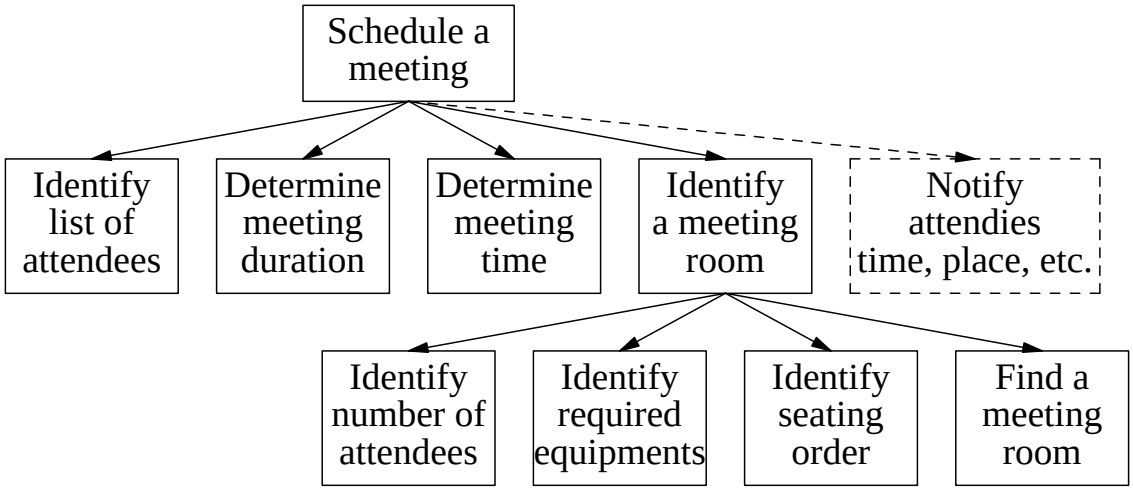
**Question:**

•? Which work-units in the above WBS are task/project specific?

•? Any two brother nodes are *mutually exclusive* - why?

# WBS-FOR SCHEDULING A MEETING

**Assume:** We know the goal of (topic to be discussed in) the meeting.




**Two Key Data Components of A Meeting:**

- The people (a kind of data!) participating/attending the meeting.
- The information/issues (also a kind of data) to be discussed or decided or distributed.

**Data-Consistecy Property:**
Each work-unit in a WBS for a project must relate to one or more data items associated with the project and vice-versa.

**Question:**

•? Why did we assume that goal of the meeting is known?

•? Does the "Why" of the meeting correspond to the objective, the goal, both, or none? Explain with appropriate example(s).

•? Show a possible breakdown of "Determine meeting duration".

•? Why "Identify seating order (arrangement)" should be a child of "Identify meeting place" instead of child of the root-node? Give an example situation where the latter might be more appropriate.

# HOW MUCH BREAKDOWN IS ENOUGH

- 80% rule: no work-unit should represents ≥ 80% of total work.
- We should be able to
  - estimate all resources needed to complete each work-unit (and hence the subtask at each intermediate node), and
  - determine the dependency relations with other work-units (which are needed for creating activity network).

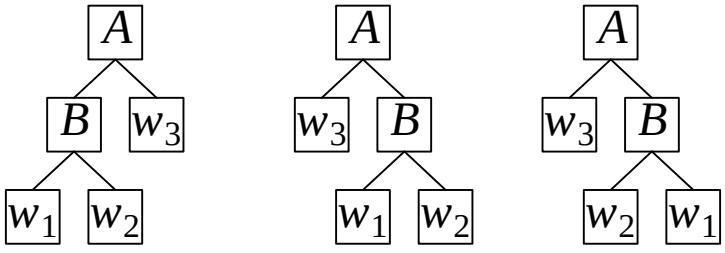
**WBS Document:**

- Give the root node a project name and number.
- Children of a node are numbered using decimal points.
- Provide explanation of terms as needed; include author and date.

**Example** (the WBS on previous page):

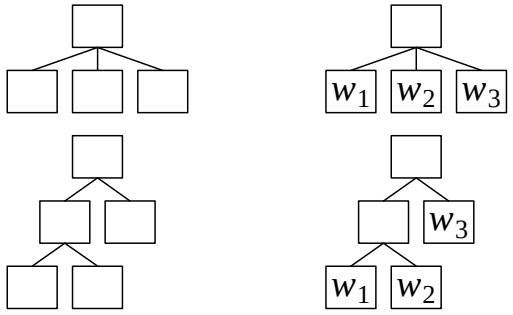
12: Research paper #12
- 12.1 Abstract
- 12.2 Body
  - 12.2.1 Introduction & Conclusion
  - 12.2.2 Defi nitions & Other texts
  - 12.2.3 Applications
    - 12.2.3.1 Appl. text
    - 12.2.3.2 Appl. fi gures
  - 12.2.4 Citations (references and fi gures)
- 12.3 References

**Question:** Are the following WBSs the same?



# SHAPES OF (UNORDERED) WBS TREE

- As unordered tree, there are only two different shapes (i.e., not distinguishable without node labels) of WBS tree with 3 terminal nodes as shown below.

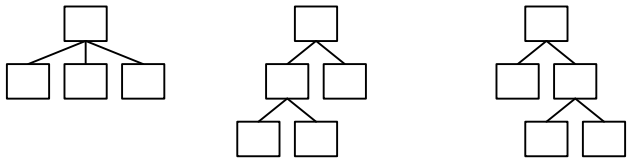



One possible labeling of the terminal nodes.



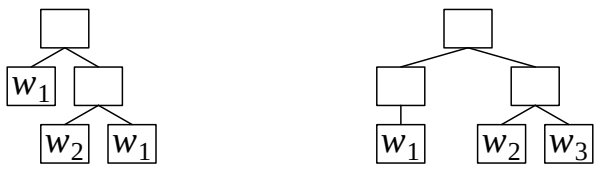

Three possible labeling of terminal nodes, giving 3 WBSs.

For ordered trees, with children of each node ordered left to right, there are 3 different shapes for the case of three work-units and 6 ways of labeling the terminal nodes in each case.



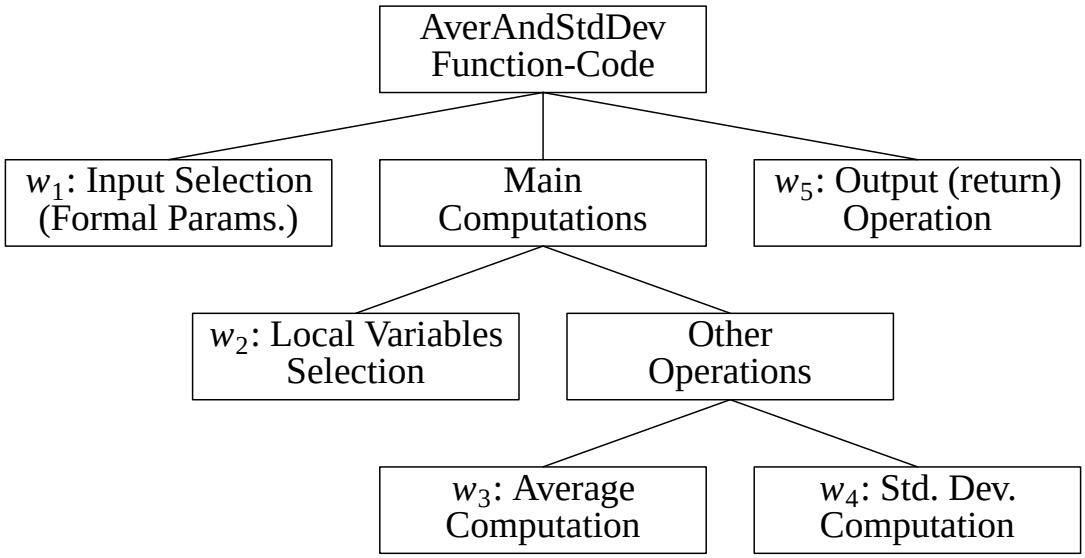
## EXERCISE

1. Show the different shapes of (unordered) WBS trees with 4 work-units $\{w_1, w_2, w_3, w_4\}$, and for each case show the total number of different WBSs.
2. Repeat Problem 1 considering the ordered trees.
3. What is wrong with the following WBSs?



## EXERCISE

1. Shown below is a WBS for creating a simple function which returns the average ($\mu$) and standard deviation ($\sigma$). The inputs consist of a data-array name and the two indicators to specify a contiguous set of array items. Assume that the inputs are error-free (hence no need for error checking/handling operations).




   (a) Give two ways of specifying the range of data-items for which we are to compute average and std. deviation.

   (b) Show a possible breakdown of $w_3$.

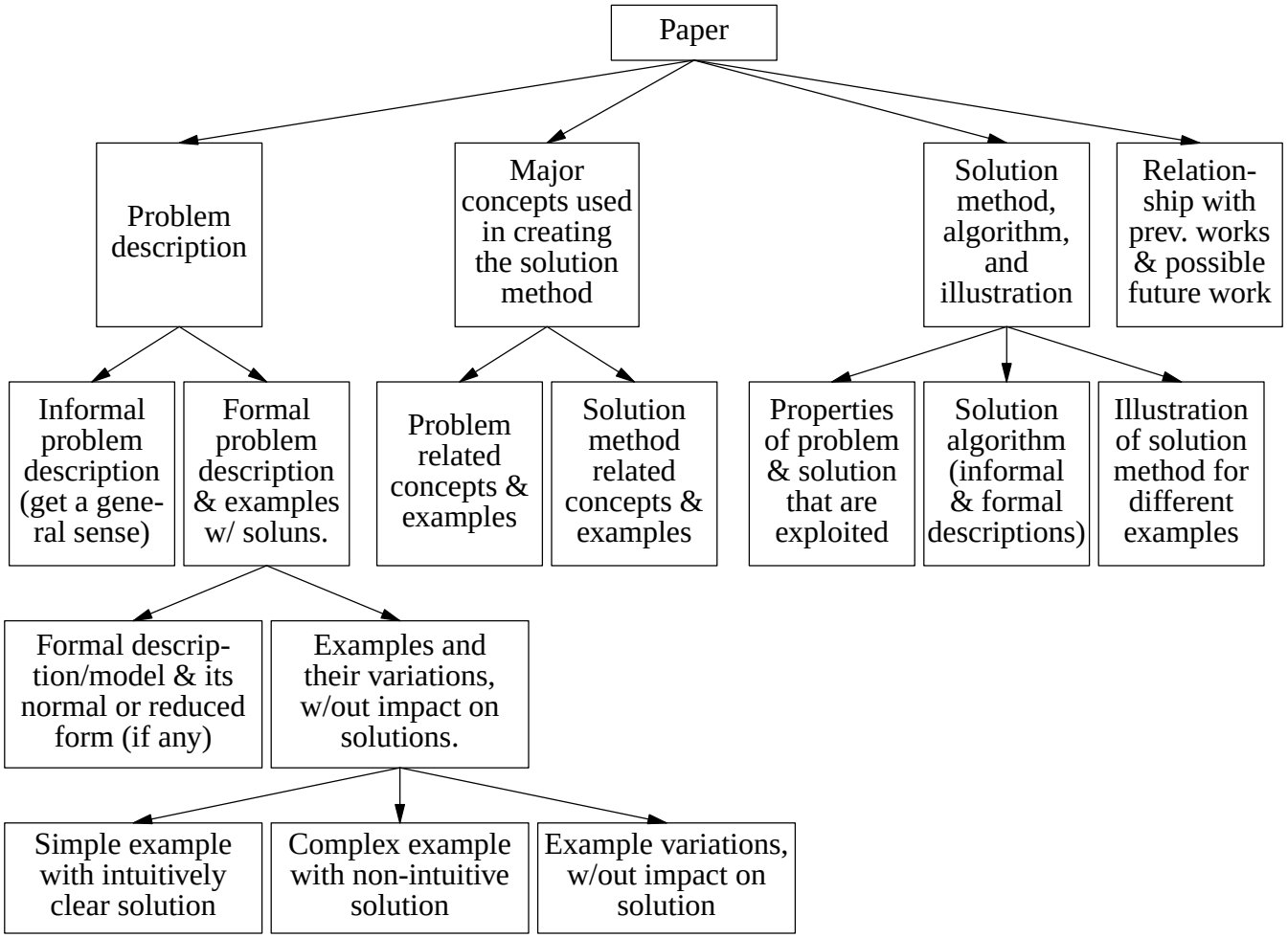
   (c) If there were global variables aver and stdDev, how would it change the WBS?

   (d) Expand the WBS with: (i) documentation operations (and its breakdowns if any) and (ii) Input-error handling.

   (e) If $w_1$ and $w_5$ in the original WBS were to be merged into one work-unit, what would be a suitable name? (Note: as we will see later, this merging is not appropriate.)

# A WBS FOR CRITICAL UNDERSTANDING OF A RESEARCH PAPER

- Read each node as "understand ..." (e.g., root node as "understand paper"). Assume that the paper addresses just one problem.
- If the paper addresses several related problems. First, identify the simplest one, and apply the following WBS; then, take up another problem which depends on just the first one, and so on.




## Question:

•? What is a possible breakdown of the node "Relationship with ···"? Are there other ways to improve this?

•? In what way does this WBS fit as you study (some aspects of) WBS itself? Indicate some content for each terminal node of the

# ACTIVITY NETWORK OF A WBS

**Task Precedence:** $w_i < w_j$ means CompTime($w_i$) ≤ StartTime($w_j$).

**Completion Dependence** (weaker than Task Precedence):

- $w_i <\bullet\, w_j$ means CompTime($w_i$) ≤ CompTime($w_j$).
- $w_i < w_j$ implies $w_i <\bullet\, w_j$.
- Though we cannot have both $w_i < w_j$ and $w_j < w_i$, we can have $w_i <\bullet\, w_j$ and $w_j <\bullet\, w_i$ giving CompTime($w_i$) = CompTime($w_j$); the start times of $w_i$ and $w_j$ may still be different.

**How Does $w_i < w_j$ Arise:**

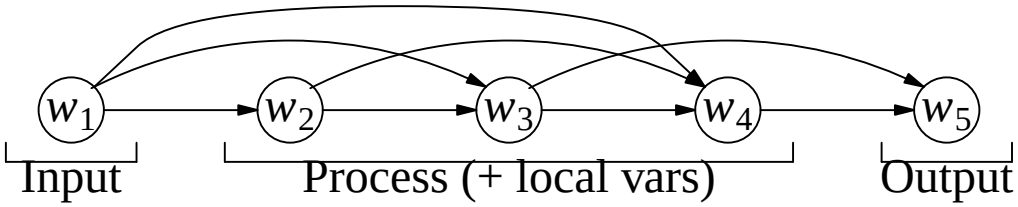
- *Primary:* One or more products/results from $w_i$ are used in $w_j$.
- *Secondary:* People/equipment used in $w_i$ will also be used in $w_j$.

**Activity Network** (AON = Activity On Nodes):

- Nodes are the work-units (activity on node - AON) $w_i$ in WBS.
- Links ($w_i$, $w_j$) are for the relation "<" ("<•" is typically not used).

**Example.** AON for the AverAndStdDev-function-code WBS before transitive reduction.

| | |
|---|---|
| $w_1$ = Input Selection<br>$w_2$ = Local Vars. Selection<br>$w_3$ = Aver. Computation | $w_4$ = Std. Dev. Computation<br>$w_5$ = Output Operation |




AON-structure and the input-process-output view;
all links here are of primary type.

## EXERCISE

1. If we decompose $w_1$ in AverAndStdDev-function-code WBS into $w_{1.1}$ (for input array-name) and $w_{1.2}$ (for two indicators for contiguous part of the input array), then show the new AON.

2. Consider a relation $R$ on work-units. It is *transitive* if $w_i \; R \; w_j$ and $w_j \; R \; w_k$ imply $w_i \; R \; w_k$ for all $w_i$, $w_j$, and $w_k$.

   Now, consider the work-units

   $w_1$: compute $x$ and $y$ (from some thing or choose randomly).
   $w_2$: compute $u = x^2 + y^2$ and $v = xy$.
   $w_3$: compute $u' = u + 2v$ and $v' = u - 2v$.

   and the following relationships on these $w_i$'s

   $w_i \; UR \; w_j$ = $w_i$ makes-use-of-the-results of $w_j$
   $w_i \; DR \; w_j$ = results of $w_i$ depends on (can be derived from) those of $w_j$.

   (Note: *DR* is based on logical/functional relationship of the data-items computed but *UR* is based on specifi c algorithms used in a computation. In particular, different algorithms for a computation may use/exploit different parts of the *DR*-relationship among the data-items involved and thus may have different *UR*.)

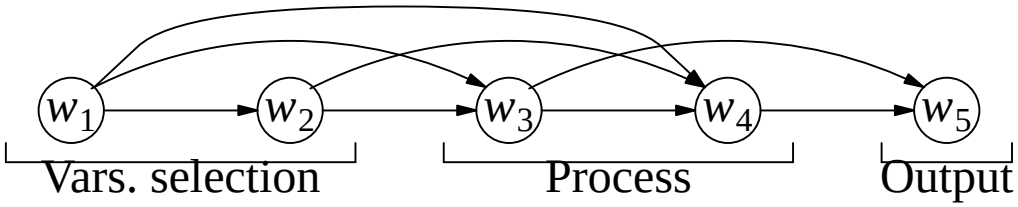
   Show the pairs $\{(w_i, w_j): w_i \; X \; w_j\}$ for $X = UR$ and for $X = DR$ using $\{w_1, w_2, w_3\}$ as above. Which of *UR* and *DR* is transitive? Show the results when we modify $w_2$ and $w_3$ by letting $u = x + y$, $v = x - y$, $u' = u^2$, and $v' = v^2$. (This is just a different algorithm for computing the values $x^2 + y^2 + 2xy$ and $x^2 + y^2 - 2xy$.)

3. Which of *UR* and *DR* is transitive in general? Which of the relationships *UR* and *DR* is stronger than the other? (For numbers, $x < y$ implies $x \le y$ and hence "<" is stronger than "≤".)

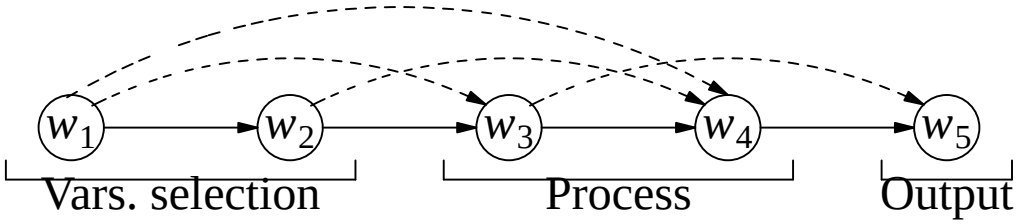
4. Which of *UR* and *DR* should be then used in an AON (and why)?

# AN ALTERNATIVE WBS FOR AverAndStdDev FUNCTION-CODE

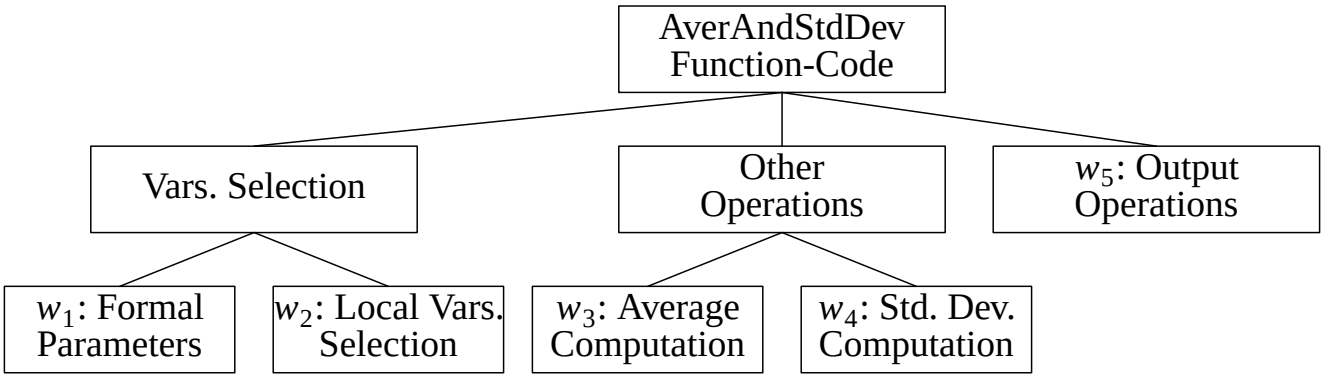
## Original AON:




## After Transitive Reduction:




## Alternative WBS:




Variables-Process-Output view of software developer vs. Input-Process-Output view of software user.

# SAMPLE PROGRAM CODES RELATED TO THE ACTIVITY NETWORK

## Code Corresponding Closely to the Activity Network:

```
01. doublePair AverAndStdDev(int *dataArray, int startIndx, int endIndx)
02. { int i;
03.   double average=0.0, //initially sum of data-items
04.          stdDev=0.0;  //initially sum of squares of data-items
05.   doublePair output;
06.   //w3
07.   for (i=startIndx; i<=endIndx, i++)
08.       average += dataArray[i];
09.   average /= (endIndx - startIndx + 1);
10.   //w4
11.   for (i=startIndx; i<=endIndx, i++)
12.       stdDev += dataArray[i] * dataArray[i];
13.   stdDev /= (endIndx - startIndx + 1);
14.   stdDev = sqrt(stdDev) - average * average;
15.   //w5
16.   output.first = average; output.second = stdDev;
17.   return(output);
18. }
```

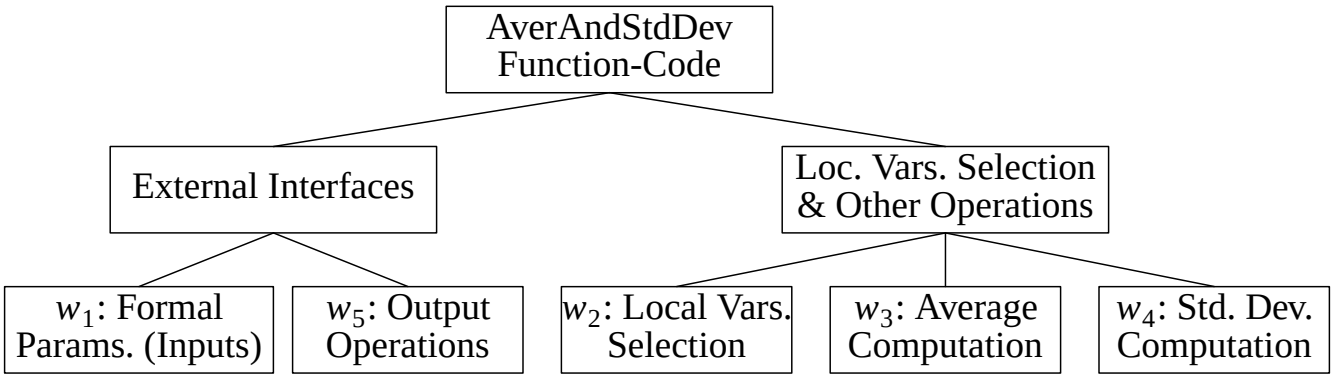
**Question:** Show the line(s) of code that justify each link in the AON.

## A Variation of Above Code (no clear separation of w3 and w4):

```
doublePair AverAndStdDev(int *dataArray, int startIndx, int endIndx)
{ int i;
  double average=0.0, //initially sum of data-items
         stdDev=0.0;  //initially sum of squares of data-items
  doublePair output;
  //w3 + w4 (w4 consists of the bold parts; added after w3 is done)
  for (i=startIndx; i<=endIndx, i++) {
      average += dataArray[i];
      stdDev += dataArray[i] * dataArray[i];
  }
  average /= (endIndx - startIndx + 1);
  stdDev /= (endIndx - startIndx + 1);
  stdDev = sqrt(stdDev) - average * average;
  //w5
  output.first = average; output.second = stdDev;
  return(output);
}
```

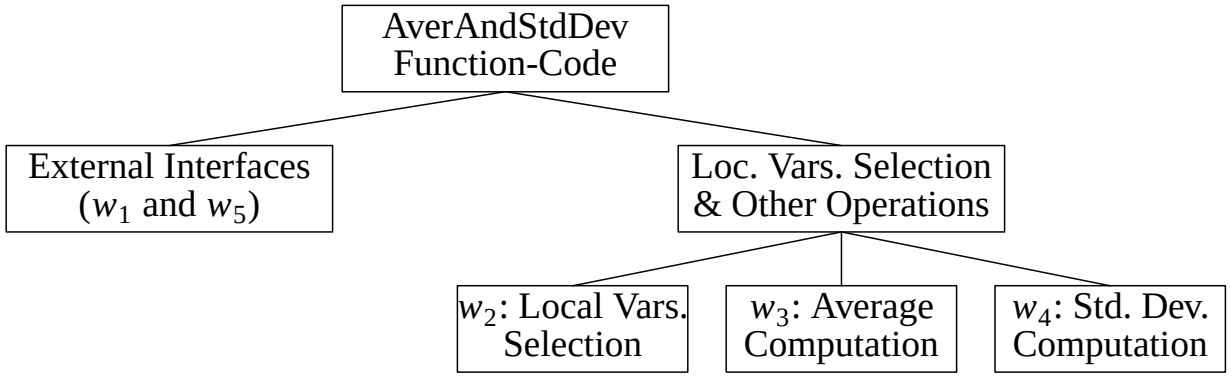
# SOME NOT SO GOOD WBS FOR AverAndStdDev-FUNCTION-CODE

## Reorganizing Same Work-Units:




- It has same AON as before (no change in the work-units).
- From the *designer viewpoint*, some may prefer to have the node "External Interface", but from AON viewpoint this is not a good idea (will not be a "proper WBS", to be defi ned shortly).
  - *Solution*: Keep input and output interfaces separated at a higher level.

**A Bad Innovation:** Merge $w_1$ and $w_5$ (creates a cycle in AON).



# WHEN TO DECOMPOSE A NODE

**Decompose:**

- Break up cycles in activity network; the new component nodes may or maynot have links between them.
- Better resource allocation.
- Better activity-monitoring by early detection of potential delays in the project completion by keeping an eye on the critical paths and putting additional resources as the need arises.
- Reorganize the WBS for better structure with respect to the activity network.

**Merging Nodes:**

- Eliminate a cycle by merging all nodes in the cycle (more generally, in the strong component containing the cycle).

  This can be done only if they have a common parent (and for that to happen we may need to reorganize the WBS hierarchy).

**Equivalent Work-Units:**

- Let $In(w_j) = \{w_i : (w_i, w_j)$ is a link in the AON$\}$.

  Likewise, let $Out(w_j) = \{w_k : (w_j, w_k)$ is a link in the AON$\}$.
- Work-units $w_i$ and $w_j$ are *link-equivalent* (in short, $w_i \approx w_j$) if $In(w_i) = In(w_j)$ and $Out(w_i) = Out(w_j)$.

**Merge Link-Equivalent Work-units:**

- This simplifi es the activity network and the WBS.

# EFFECT ON SCHEDULES FROM MERGING EQUIVALENT WORK-UNITS

**Equivalent Work-Units:** $w_i$ and $w_j$.

**A Schedule:**

$$\cdots, w_i, \underline{w_{k_1}, w_{k_2}, \cdots, w_{k_n}}, w_j, \cdots$$

**New Schedules:** $w_{ij}$ is the merged work-unit.

- The part between $w_i$ and $w_j$ can be split into two parts ($0 \leq m \leq n$), with the left part moved before $w_i$ and the right part moved after $w_j$, as shown below.

$$\cdots, \underline{w_{k_1}, \cdots, w_{k_m}}, w_{ij}, \underline{w_{k_{m+1}}, \cdots, w_{k_n}}, \cdots$$

# THREE KEY PROPERTIES OF AON

## Acyclicity

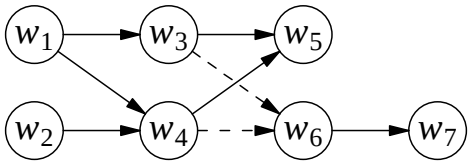
- There should not be any cycle in an AON.

## No Transitive Links:

- There should be any transitive link $(w_i, w_k)$ if there are links $(w_i, w_j)$ and $(w_j, w_k)$.
- More generally, if there are links $(w_{i_1}, w_{i_2})$, $(w_{i_2}, w_{i_3})$, $\cdots$, $(w_{i_k}, w_{i_{k+1}})$, then the link $(w_{i_1}, w_{i_{k+1}})$ should not be there.

**Connectedness** (as an undirected graph):

- Deletion of the dashed links below splits the AON into two disconnected parts $\{w_1, w_2, \cdots, w_5\}$ and $\{w_6, w_7\}$, and we can consider this to represent two disjoint projects with the above sets of work-units for them.

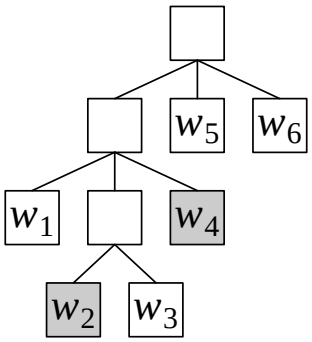



## Question:

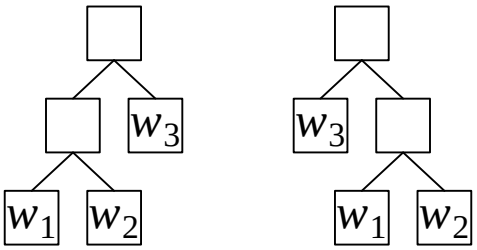
•? List all implicit or explicit precedence constraints among the work-units in the above AON that have to be removed in order to be able to delete the links $(w_3, w_6)$ and $(w_5, w_6)$?

## EXERCISE

1. If we have to merge the terminal nodes $w_2$ and $w_4$ in the WBS below, then some other non-terminal nodes have to be also merged. What are they, and show the new WBS after the merging.




2. If we know the precedence $w_1 < w_3$, then which of the following representation of the WBS is preferred?



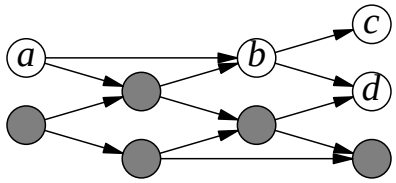

3. If we know $w_1 < w_2 < w_3 < w_4$, show all possible proper representations of a WBS on $\{w_1, w_2, w_3, w_4\}$.

# PATH-CLOSED PROPERTY AND A PROPER WBS

## Path-Closed Property:

- A subset of nodes $S$ in an activity network is *path-closed* if for each $w_i$, $w_j \in S$ and each path $\pi_{i,j}$ from $w_i$ to $w_j$, $\pi_{i,j} \subseteq S$.
- The presence or absence of transitive does not affect path-closedness of $S$.

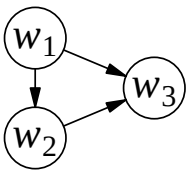
**Example.** The dark nodes in the AON below form a path-closed subset; the complement set $\{a, b, c, d\}$ does not (why?).



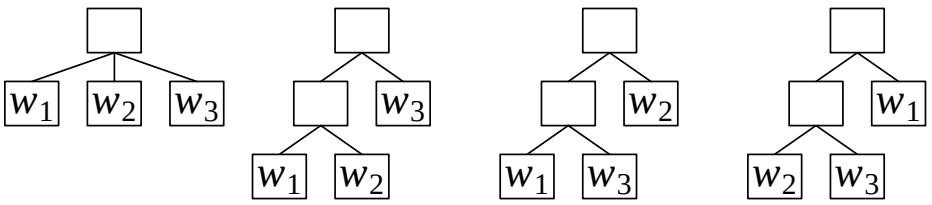

## Proper WBS:

- For each non-terminal node $x \in$ WBS, the set $W(x) = \{w_i: w_i$ is a descendent of $x\}$ is connected and path-closed.
- The activity network is weakly connected, i.e., for any two $w_i$ and $w_j \neq w_i$, there is an undirected path connecting them.

**Example.** There are four possible WBS with 3 terminal nodes; all are proper if we delete link $(w_1, w_2)$ in (i).




(i) A small activity network.




(ii) The first, second, and fourth are proper; the 2nd and 4th maybe considered better than 1st.

## EXERCISE

1. If $S_1$ and $S_2$ two path-closed sets, then $S_1 \cap S_2$ is also path-closed (if it is not empty).
2. For any $w_i$ and $w_j$, show that the set $W(w_i, w_j) = \{w_k: w_k$ is on some path from $w_i$ to $w_j\}$ is the smallest path-closed set containing $\{w_i, w_j\}$ if it is not empty.
3. Suppose $\langle w_1, w_2, \cdots, w_N\rangle$ is an arbitrary topological ordering of an AON. Show that each $W_{i,j} = \langle w_i, w_{i+1}, \cdots, w_j\rangle$, $1 \le i \le j \le N$, is a path-closed set. (Thus, there is a proper WBS of each different shape; we just need to assign $w_i$'s properly to the terminal nodes; see the example above.)
4. Defi ne the transitive closure $\bar{N}$ of an activity network $N$ by
   - $\bar{N}$ has the same nodes $w_i$ as $N$ (including the required time $R(w_i)$ for completion of $w_i$).
   - There is a link $(w_i, w_j) \in \bar{N}$ iff there is a path from $w_i$ to $w_j$.

   Prove that a subset of nodes $S$ is path-closed in $N$ if and only it is path-closed in $\bar{N}$. (In fact, for any $N'$ such that $N \subseteq N' \subseteq \bar{N}$, $N'$ has the same path-closed subsets $S$ as $N$. Also, the critical paths, schedules, and milestones are the same for $N$, $N'$, and $\bar{N}$. This shows that we can ignore transitive links in analysis of AON.)
5. How would you generate a random AON (not necessarily, weakly connected) on a given number of nodes so that each link has the same probability of $p$ being chosen. How will the algorithm change if we want the network to be free of transitive links (or to be weakly connected)?
6. If $S_1 \subset S_2$ and $S_1$ is path-closed in the subnetwork $N(S_2)$ of $N$ restricted to $S_2$. Show that (1) if $S_2$ is path-closed in $N$, then $S_1$ is path-closed in $N$, and (2) if $S_2$ is not path-closed in $N$, then $S_1$ need not be path-closed in $N$. (This makes testing the path-closedness of $S_1$ easier if we know a path-closed $S_2 \supset S_1$.)

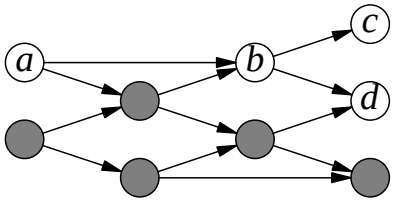
# TESTING PATH-CLOSEDNESS

**Algorithm:** Complexity $O(|V| + |E|)$.

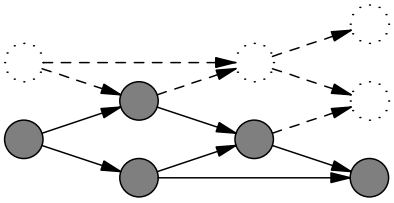
*Input:* An acyclic digraph $\vec{G} = (V, E)$ and a node-subset $S \subseteq V$.
*Output:* Yes/no.

1. Successively delete nodes $y \notin S$ with indeg($y$) = 0 or outdeg($y$) = 0. In the first case, delete all links ($y$, $z$) from $y$ and reduce each indeg($z$) by 1; in the second case, delete all links ($x$, $y$) to $y$ and reduce each outdeg($x$) by 1.
2. If there are no nodes left outside $S$, then $S$ is path-closed, else $S$ is not path-closed.

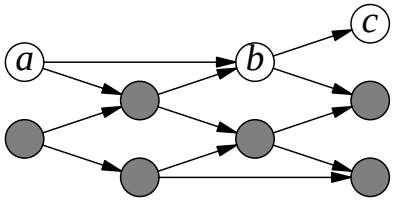
**Example.** Top: the dark nodes form a path-closed subset. Bottom: the dark nodes *do not* form a path-closed subset.



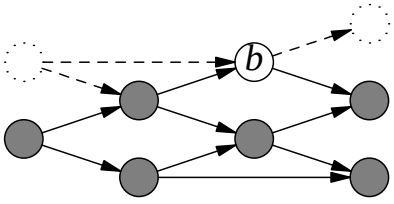

The nodes $a$, $c$, $d$, $b$ can be deleted in that order.



After deletion of $\{a, c, d, b\}$.




The nodes $a$ and $c$ can be deleted but not node $b$.




After deletion of $\{a, c\}$.

**Question:** Generalize the algorithm when there are cycles.

# A PARTIAL ORDERING ON THE NODES OF WBS

## Ordering Two Arbitrary Nodes:

- Consider two nodes $x$ and $y$ such that neither is a descendent of the other.
- If for some $w_i \in W(x)$ and some $w_j \in W(y)$ we have $w_i < w_j$, then we say $x < y$.

## Question:

•? Prove that $x < y$ implies that $y \not< x$.

•? Prove that if we merge the nodes in $W(x_i)$ for one or more non-terminal nodes $x_i$ in a proper WBS (we may assume that no two $x_i$ have descendent-ancestor relation between them), then the resulting AON is still acyclic.

## Convention on Arranging Children of A Node:

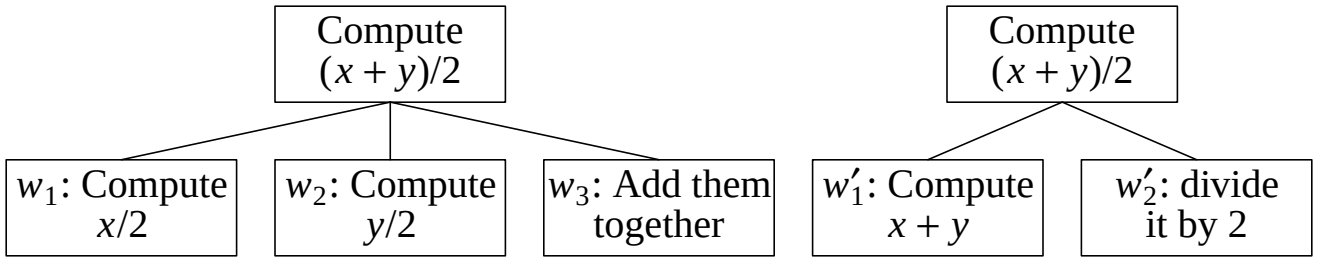
- Order the children of each node in WBS in the left to right order consistent with "<".

# WHAT MAKES CREATING A WBS A NON-TRIVIAL TASK

**Different WBS for The Same Task:**

- Two WBS at the same level of detail may have a different set of work-units (and hence perhaps a different activity network).
  - The choice of the work-units represents the steps of a specific *solution approach* (but possibly not of a solution algorithm) for the original task.
  - This is what makes creating a WBS a non-trivial task.
  - The AON is determined by the work-units; it represents a less abstract form of a solution algorithm than the WBS.
- The notion of path-closed subsets helps us to select a better WBS among the many possible WBS having the same AON.

**Example.** Shown here are two different WBS for the task "Compute the average of $x$ and $y$" (no work-unit for choosing inputs.) In the first case, there are two algorithms (what are they?). The two WBSs give different algorithms for computing $(x + y)/2$.




**Question:**

•? Why is it that a WBS may not truly represent a software, even if the AON is linear (after transitive reduction)?

# EXERCISE

1. Consider the WBS in Fig. 1 below.
   (a) Show an appropriate activity network and use it for analyzing the goodness of the WBS.
   (b) What would justify decomposing the node "List lecture goals/summary" in Fig. 1 as shown in Fig. 2? Does it suggest useful decompositions of $w_4 - w_5$ and $w_7 - w_9$; show such decompositions.
   (c) Should we reorganize WBS in Fig. 1 by distributing these decomposed parts under "List Prob #1 solved" and "List Prob #2 solved"?

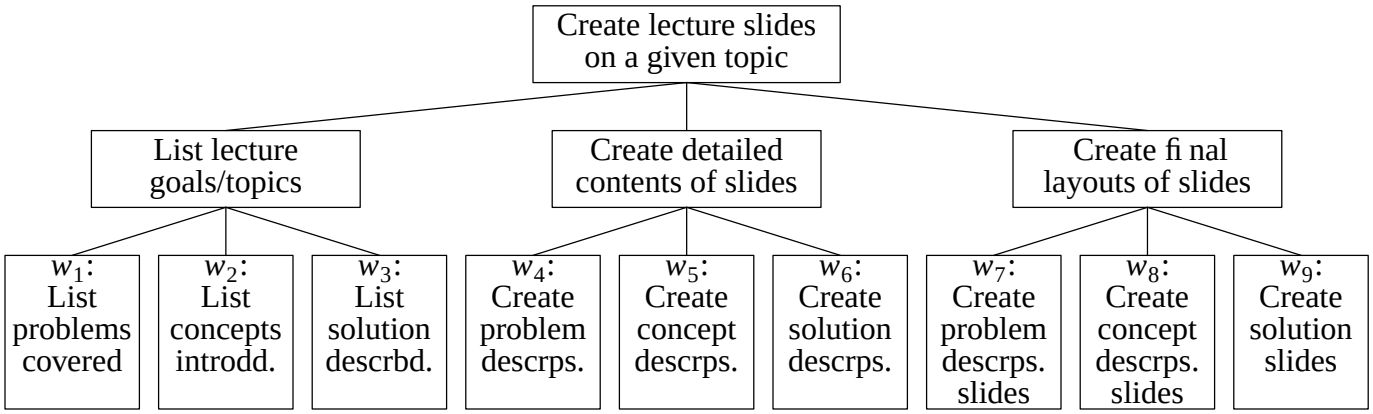



Figure 1. A suggested WBS for the task "Create lecture slides."

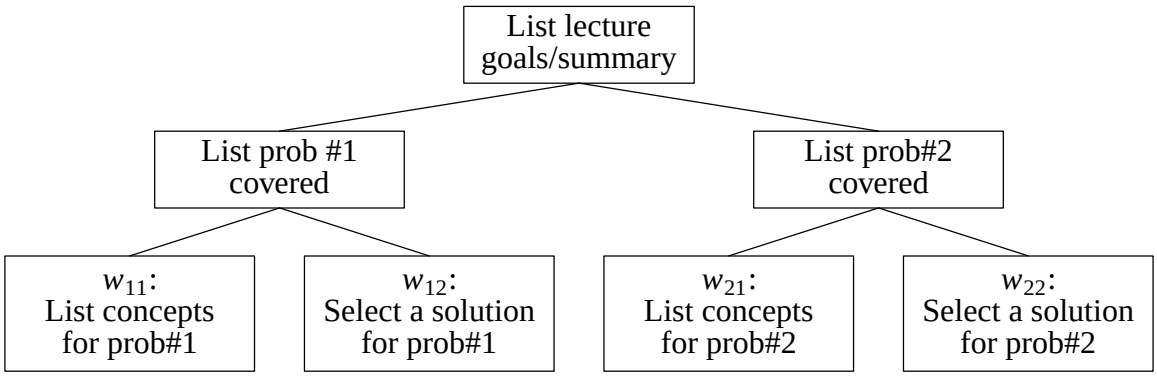



Figure 2. An alternate decomposition of the task "List lecture goals/summary."

## EXERCISE (Contd.)

2. Shown below is a WBS that combines the ideas from both Fig. 1 and Fig. 2. In what way this is better/worse than both of those?

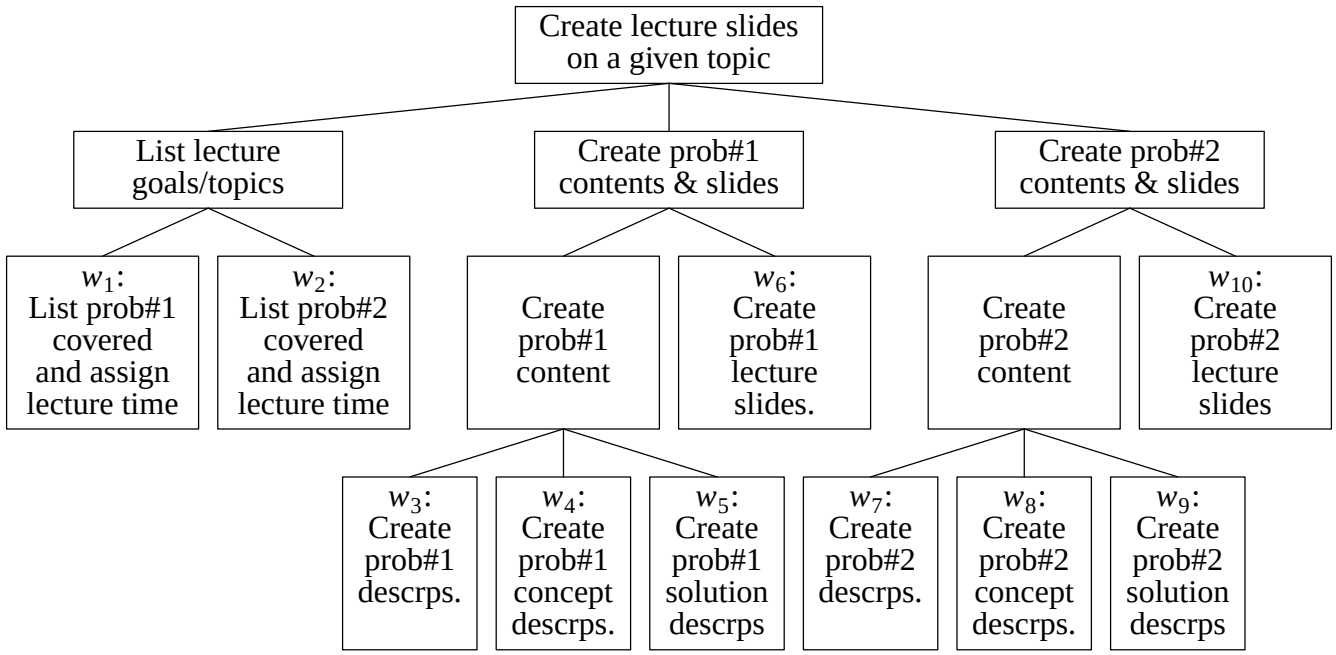
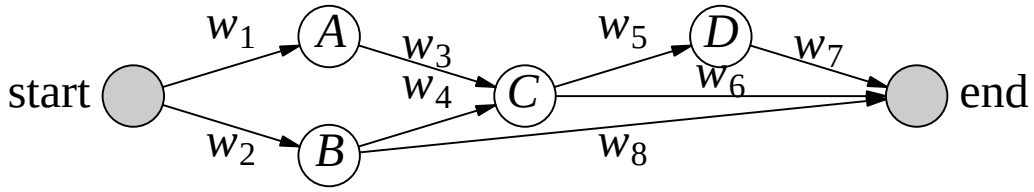



Figure 3. A WBS which combines ideas from Figs. 1 and 2.

3. Is it worthwhile to improve the task-descriptions for $w_3$ to $w_5$ and $w_7$ to $w_9$ in the WBS in Fig. 3 by adding "allocate time" for each of them? Justify yor answer.

4. Show the activity network for WBS in Fig. 3. Is there any good reason to merge $w_1$ and $w_2$ (because the sum of lecture-time allocated to Prob#1 and Prob#2 must add to the total lecture time = 50 minutes, say), and if so show the new WBS and the description of the merged task $w_{12}$ (without loss of information). If we must keep $w_1$ and $w_2$ separate, then how do we change these tasks (you can introduce another task or work-unit, if needed)? Is there any need to do similar modifi cations for the other parts of the WBS?

# ACTIVITY NETWORK WITH ACTIVITY-ON-LINK

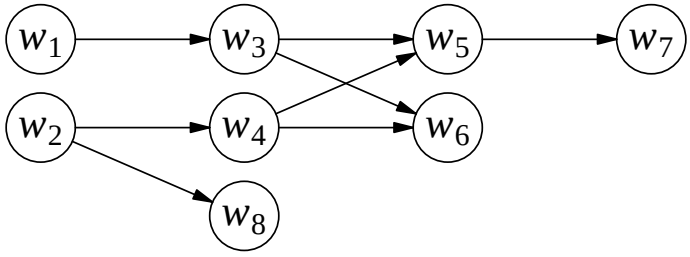



Nodes = states of the project.

**Question:**

- •? What does node *C* here say about the project's state?
- •? Why is it that we can always assume there is a unique source-node and a unique sink-node (unlike an AON)?
- •? Can we assume that there is no transitive link?

**Conversion to Standard AON Form:** Nodes for work-units.




**Algorithm for AOL-to-AON Conversion:**

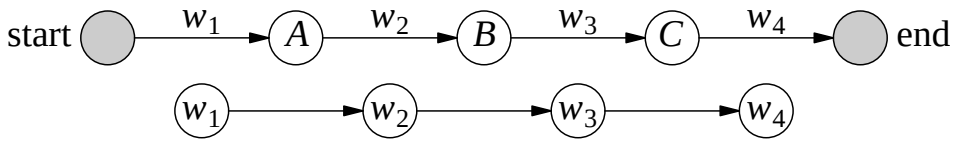
1. Create a node for each $w_i$.
2. Add the link $(w_i, w_j)$ to AON if in the AOL the head of the link $w_i$ equals the tail of the link $w_j$.

**Question:**

- •? Formulate a recursive form of the algorithm (based on deletion of an appropriate link to reduce the problem size).

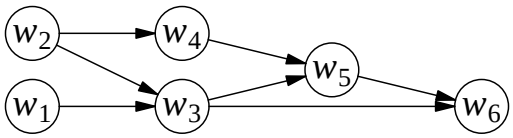
# LIMITATIONS OF THE ACTIVITY-ON-LINK FORM

## Both Forms Work Well for the Simplest Networks:




## AOL-Representation Does not Always Work:

- The following AOL has no AOL-representation. (Removing the transitive link ($w_3$, $w_6$) is of no help.)




## Question:

•? What are some minimal changes (addition or deletion of some links) to the AON that will allow a AOL-representation?

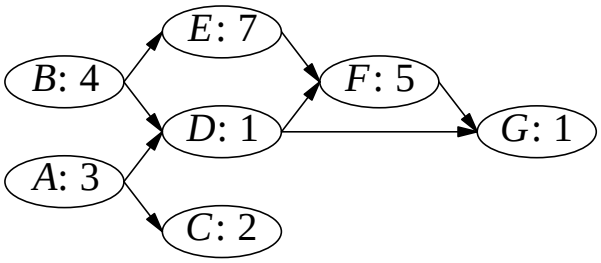
•? Give a necessary and suffi cient condition for an AON being representable by an AOL. Does this condition imply that the AON has no transitive link?

## Algorithm for AON-to-AOL Conversion:

1. Remove a source-node $w_i$ from the AON and obtain an AOL representation for the reduced AON.
2. Add a new node to the AOL and add a link from this node to another node in the AOL to accommodate the links from $w_i$ in AON; label the new link in AOL $w_i$.

**Note:** Step (2) cannot be always done; when can we do it?

# CRITICAL-PATH ANALYSIS

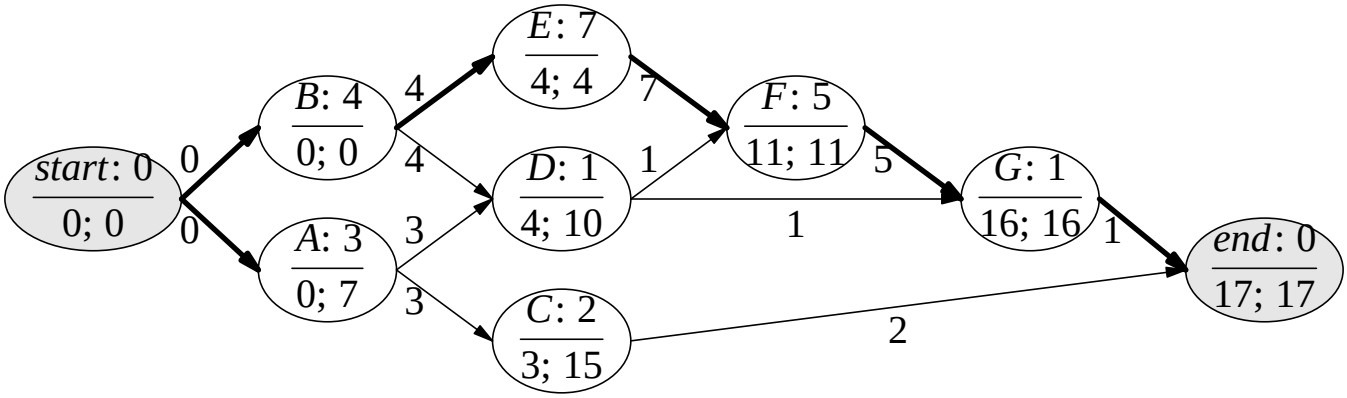



- The number next to each node is the time to complete it.
- $(D, G)$ is a transitive link.

**Earliest and Latest Start-Times** (without causing project-delay):

- $R(w_i)$ = Time required to complete the work $w_i$.
- $E(w_i)$ = The earliest time to start $w_i$.
- $L(w_i)$ = The latest start-time of $w_i$.
- $S(w_i)$ = The slack-time for $w_i = L(w_i) - E(w_i)$.

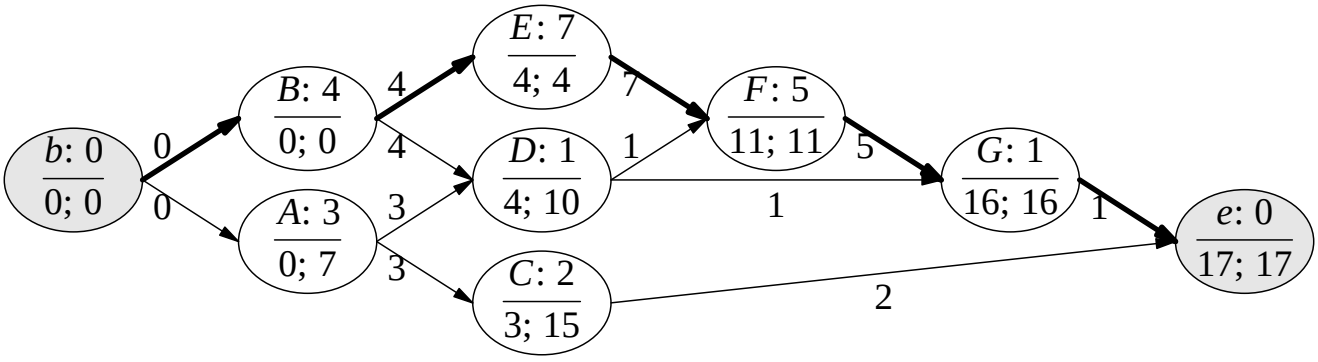
**Critical-Path:** A longest-path from "start" to "end".



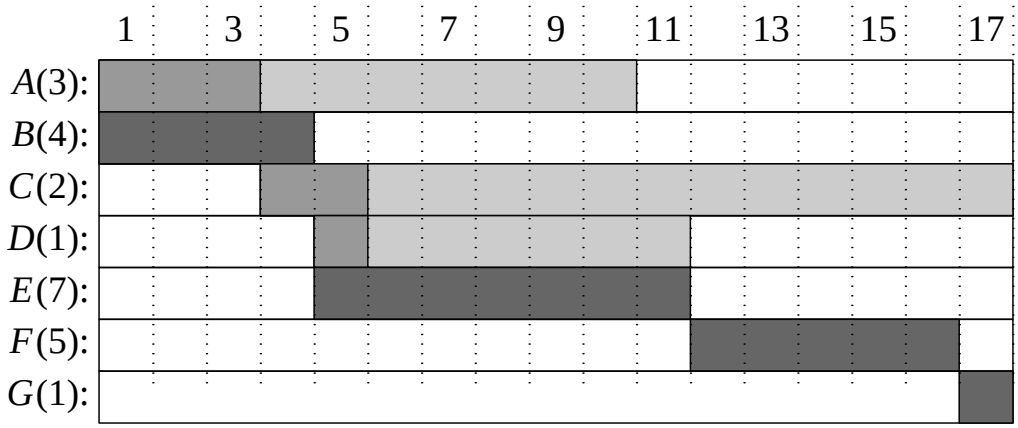

The two numbers below the bar at each node are $E(w_i)$ and $L(w_i)$.
$L(D) = \min\{L(F), L(G)\} - R(D) = \min\{11, 16\} - 1 = 10$.

- All tasks $\{B, E, F, G\}$ on critical path(s) have $S(w_i) = 0$; any delay in their completion will cause the whole project delayed.
- The presence of transitive link does not affect the critical path, but it increases computation time of $E(w_i)$ and $L(w_i)$.
- If $(w_i, w_j)$ is a critical-link, i.e., $E(w_i) + R(w_i) = E(w_j)$, then it is not a transitive link. (Links on critical paths are critical links.)

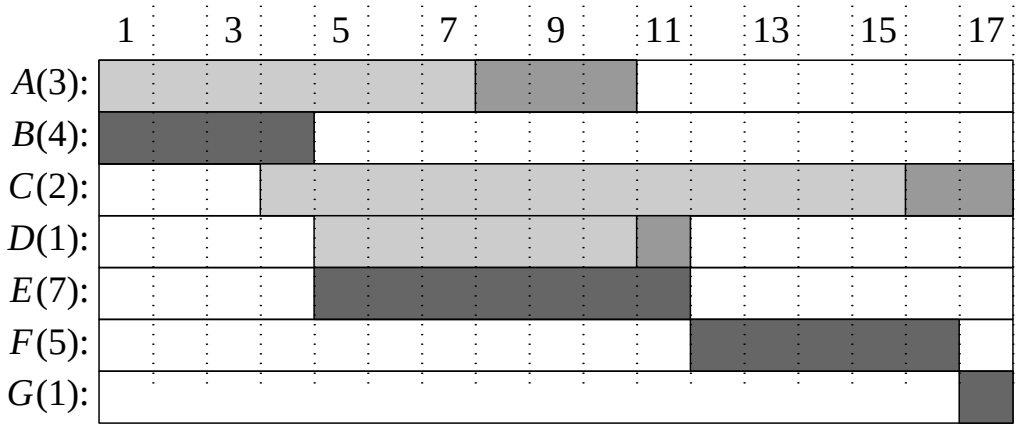
# GANTT CHART




- To reduce project duration, we must reduce $R(w_i)$ for some $w_i$ on each critical path, by allocating more resources (man-power, etc).

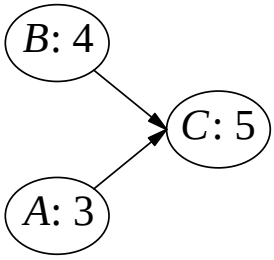



(i) Scheduling based on $E(w_i)$, without using slack-time; degree of parallelism is 3, among $\{C, D, E\}$.

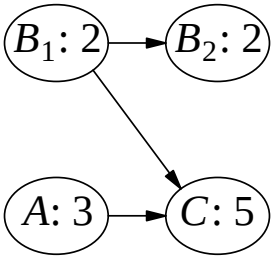



(ii) A better scheduling based on $L(w_i)$, with full use of slack-time; the lower degree of parallelism 2 puts less demand on manpower.

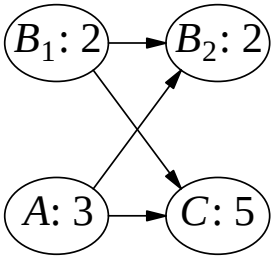
# A MORE DETAILED WORK-BREAKDOWN MAY LEAD TO A BETTER SCHEDULE



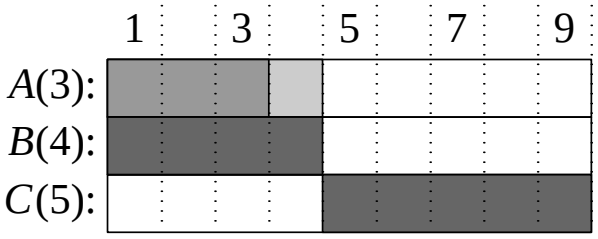

(i) A simple AON.



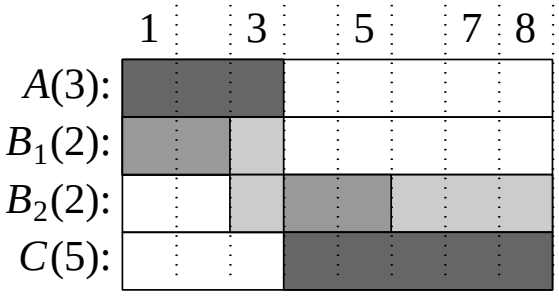

(ii) A possible result of breaking $B$ into $B_1$ and $B_2$.




(iii) This is not possible (why?).




(iv) An optimal schedule for (i) based on $E(w_i)$.

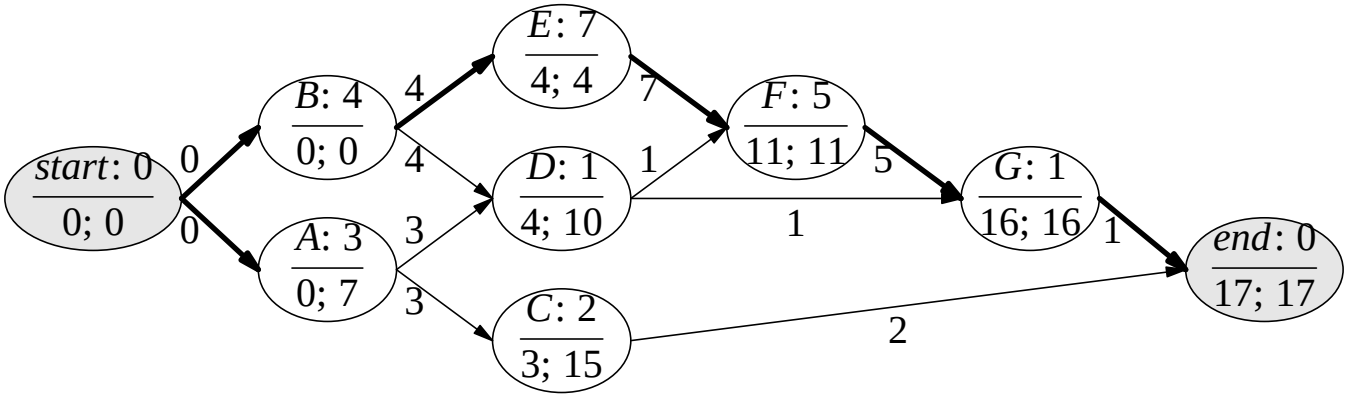



(v) An optimal schedule for (ii) based on $E(w_i)$ and keeping the parallelism = 2 as above.

**Question:**

- •? Show all other activity networks that can result due to breaking of $B$:4 into $B_1$:2 and $B_2$:2. Show their optimal schedules. Which one does not reduce the completion time?
- •? Can a more detailed work-breadown ever lead to an worse optimal schedule? If so, give an example.
- •? Why is it that a breakdown of $A$ (without that of $B$) does not help to reduce the completion time?
- •? Will a breadown of $C$ reduce the completion time?

# COMPUTING $E(w_j)$ AND $L(w_j)$




The two numbers below the bar at each node are $E(w_i)$ and $L(w_i)$.

**Property of $E(w_j)$:**

- $E(w_j) = \begin{cases} 0, \text{ if there are no links } (w_i, w_j) \text{ to } w_j, \text{ and otherwise} \\ \max \{E(w_i) + R(w_i): (w_i, w_j) \text{ is a link in AON}\} \end{cases}$

**Example:**

- $E(D) = \max \{E(A) + R(A), E(B) + R(B)\} = \{0+3, 0+4\} = 4.$
  $E(G) = \max \{E(D) + R(D), E(F) + R(F)\} = \{11+5, 4+1\} = 16.$

**Top-Down Computation of $E(w_j)$'s:**

- $E(w_j)$ must be computed in top-down (left to right) fashion, starting with nodes that have no incoming link. (Acyclicity of AON assures that there is always such a node.)

**Question:**

•? State a similar property of $L(w_j)$, and give the computation method for $L(w_j)$'s. How does it differ from that of $E(w_j)$'s?

•? Call a link $(w_i, w_j)$ *super-critical* if $L(w_j) = E(w_i) + R(w_i)$. State the connection among critical paths, critical links, and super-critical links.

•? What is the maximum $R(D)$ possible for project duration 17?

•? How does node merging impact critical paths and project-duration?

# A BRIEF REVIEW OF WBS

- Create initial WBS to identify work-units.
- Identify the precedence relationships and determine AON.
- First Analysis of AON:
  - Check acyclicity (and modify WBS, if necessary, by decomposing or merging nodes).
  - Reorganize the structure and/or labeling of intermediate nodes of WBS (not work-units) so that each $W(x)$ is path-closed.
  - Make WBS an ordered-tree based on the relationship "$x < y$" for all (terminal and non-terminal) nodes.
- Second Analysis of AON:
  - Find $R(w_i)$, $E(w_i)$, $L(w_i)$, and the critical paths.
  - Find schedules and milestones.
  - (Yet to cover) Decompose work-units, if needed, to provide more milestones and better distribution of workloads.

**Questions:**

•? A work-unit $w_j$ is called critical if $E(w_j) = L(w_j)$. Justify your yes/no answer for each of the following.

- For each link $(w_i, w_j)$ in AON, $E(w_i) + R(w_i) \leq E(w_j)$.
- For each link $(w_i, w_j)$ in AON, $L(w_i) + R(w_i) \leq L(w_j)$.
- If $w_j$ is critical then there is at least one link $(w_i, w_j)$ such that $w_i$ is also critical.
- If $w_i$ is critical then there is at least one link $(w_i, w_j)$ such that $w_j$ is also critical.
- If both $w_i$ and $w_j$ are critical and $(w_i, w_j)$ is a link in AON (no transitive link), then $E(w_i) + R(w_i) = E(w_j)$ and $L(w_i) + R(w_i) = L(w_j)$.

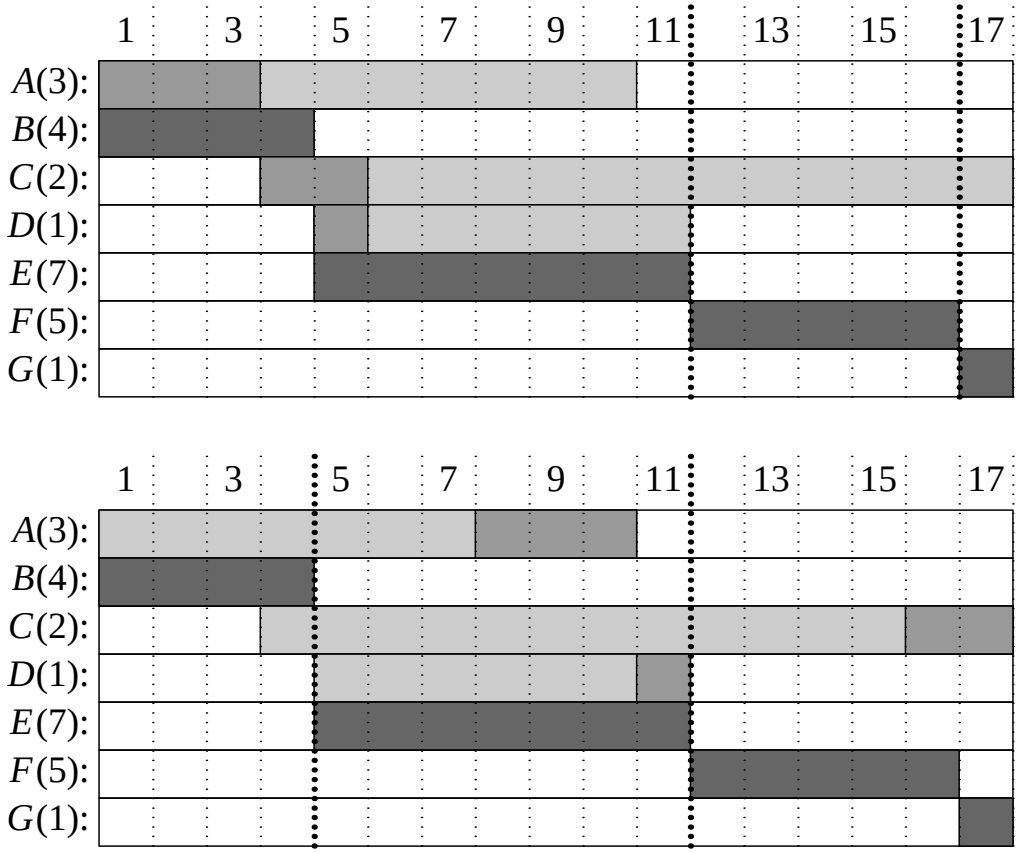
# MILESTONES: SPECIAL TIME POINTS IN A SCHEDULE

**Milestone $t_i$:** Each task which starts before $t_i$ ends before $t_i$.

**Better Schedule:**

- Minimize parallelism-degree, without increasing total duration.
- Reduce the impact of errors on project-completion time by minimizing the Milestone Distribution Measure $MDM = \sum(t_{i+1} - t_i)^2$.

**Example.** The second schedule has a lower *MDM* and is better.



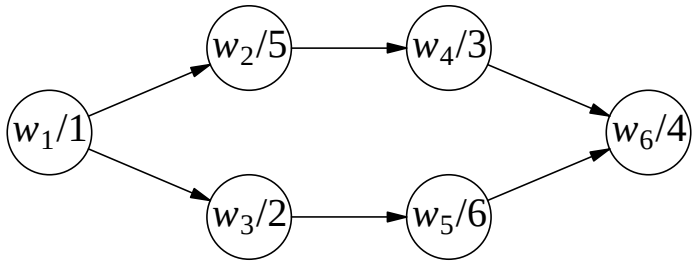

**Question:**

•? Modify the second schedule to lower *MDM* even more.

•? Obtain an algorithm to create a schedule with minimum *MDM*.

## EXERCISE

1. For the AON below, find the critical paths and the milestones (use the best schedule that minimizes MDM, without increasing project duration). Shown next to each node is $R(w_i)$.




   (a) How do these things change if we add the link $(w_2, w_5)$?

   (b) How about if we add the link $(w_3, w_4)$?

   (c) How about if we add both the links $(w_2, w_3)$ and $(w_3, w_4)$?

   (d) Why shouldn't we merge $w_2$ and $w_3$ into a single work-unit $w_{23}$? What would go wrong if we did?

   (e) How about merging $w_2$ and $w_4$ into a single work-unit $w_{24}$?

2. If we want to break up a work-unit $w_i$ in Problem 1 into 2 pieces $w_{i.1}$ and $w_{i.2}$, which $w_i$ would it be in order to provide maximum advantage in some sense (and what would be the best break-up)?

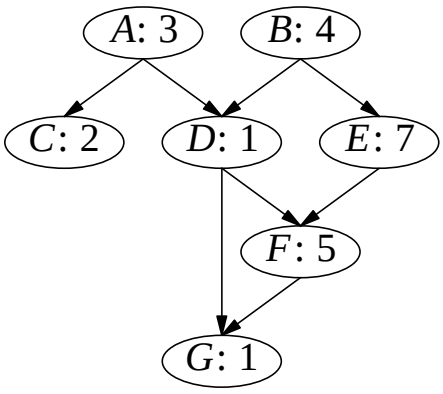
3. Give a necessary and sufficient condition for a unique schedule.

4. Is it reasonable to require that the set of work-units in a milestone range equals the union of one or more $W(x)$'s? How about requiring that each $W(x)$ equals the union of work-units in one or more milestone-ranges?

5. What do the observations in Problem 4 say about a proper WBS for the original AON?

6. Argue that if $w_i$ and $w_j$ are in a milestone's range, then all the work-units in $W(w_i, w_j)$ are in that milestone's range.

7. Modify $R(D)$ in the AON below, without increasing the length of the critical path, to show that the scheduling based on $L(w_i)$ need not be better than that based on $E(w_i)$ in terms of MDM. What is the maximum possible value of $R(D)$?

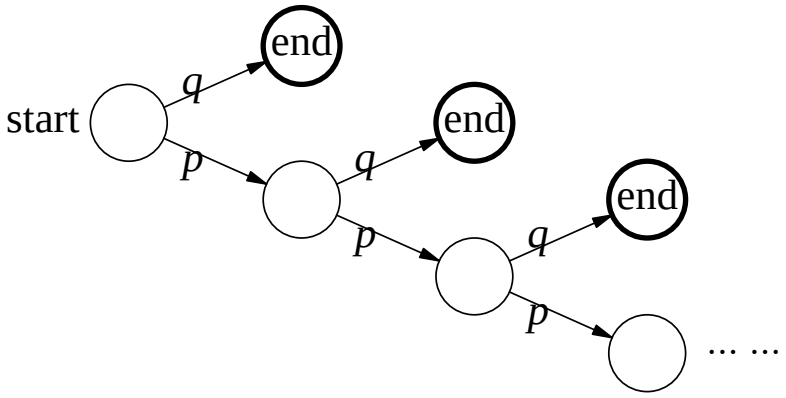



The number next to each node is the time to complete it.

# NUMBER OF ATTEMPTS NEEDED TO DO SOMETHING RIGHT

**Error Probability:** $p > 0$.

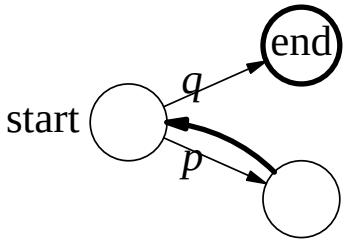
- Probability of no error = $1 - p = q < 1$.




**Average #(attempts to successful completion):**

- $E = 1.q + 2.pq + 3.p^2q + \cdots = q(1 + 2p + 3p^2 + \cdots)$
  $= q(1 - p)^{-2} = qq^{-2} = 1/q$.
- For $q = 0.1$, $E = 10$.

**Alternate Approach to Derivation of $E$:**



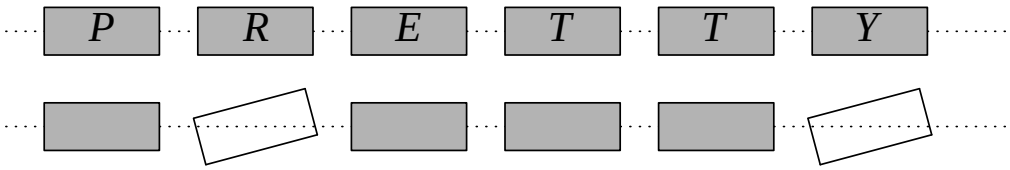

- $E = 1.q + p(1 + E) = q + p + p.E = 1 + p.E$, hence $E = 1/q$.

**Question:** How to determine $E$ for a multi-step task?

# TWO KINDS OF ERRORS

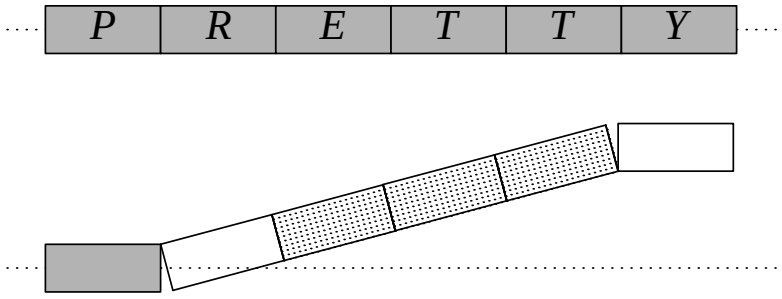
**Non-Propagating Error:** loosely-coupled case.

- Independent placement of brick (not of same size perhaps).
- Error (rotation) is localized; simple error-impact.




**Propagating Error:** tightly-coupled case.

- Closely aligned (touching) placement of *n* bricks of equal width.
- Error is not localized; complex error-impact.
- All bricks starting with the first error have to be reworked to correct the problem.




**Question:**

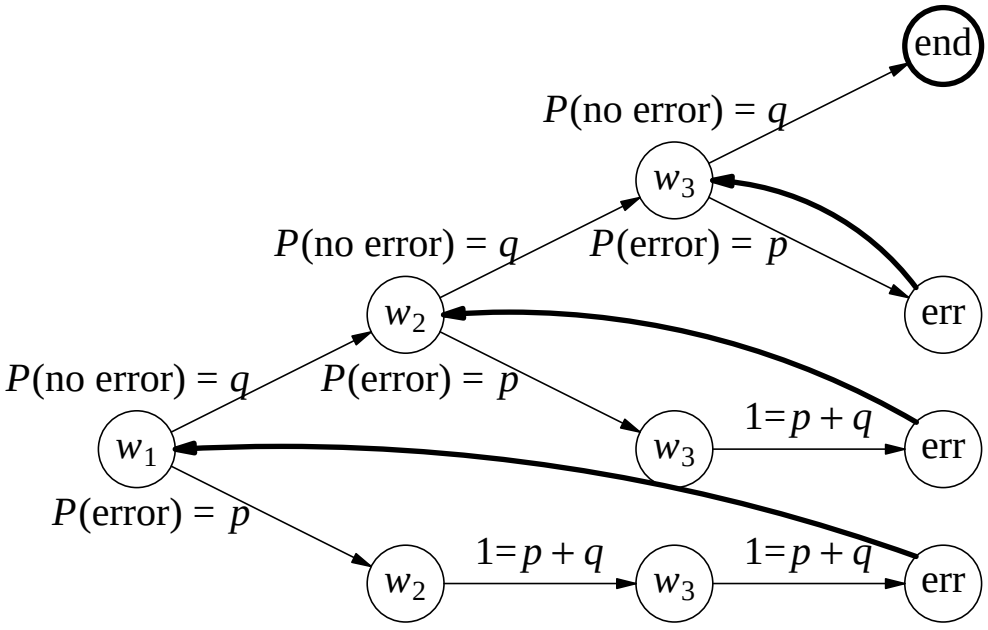
- •? Can we detect (or identify) an error from the last brick's position?
- •? Assume the only error is a rotation by $\pm\theta$. Which error-patterns put the *n*th brick back on the line? Give a condition on brick's length/width so that the *n*th brick will still be in its proper place.
- •? Give an example of assembly of some kind of pieces so that every thing has to be redone if any error occurs.

# THE MEASURE *MDM*

## A Simplistic Model:

- An work-unit $w$ of length $n = R(w)$ is broken into $n$ pieces of unit-effort (time) $w_i$, $1 \le i \le n$.
- Probability $0 < p \le 1$ for error at any of $w_i$.
- Inspection for errors occurs only at the end of $w_n$.
- The effect of error at $w_i$ is not overshadowed by any $w_j$, $j > i$.
- If the first error occurs at $w_i$, then we must rework each $w_j$, $j \ge i$.

**Average duration for error-free completion of** $w$**:** $E_n$.




Alternative execution-paths of $w_i$'s for $n = 3$.

**Case** $n$**=3:** $E_3 = p[3 + E_3] + qp[3 + E_2] + q^2 p[3 + E_1] + q^3 . 3$

**General Case:** Prob(1st error at $w_i$) = $q^{i-1} p$.

- $E_n = p[n + E_n] + qp[n + E_{n-1}] + q^2 p[n + E_{n-2}] + \cdots$
  $+ q^{n-1} p[n + E_1] + q^n [n]$.

# SOLVING THE EQUATION FOR $E_n$

**Equation for $E_n$:**

- $E_n = p[n + E_n] + qp[n + E_{n-1}] + q^2 p[n + E_{n-2}] + \cdots + q^{n-1} p[n + E_1] + q^n[n]$.

**Solving Successively $E_1, E_2, \cdots, E_n$:**

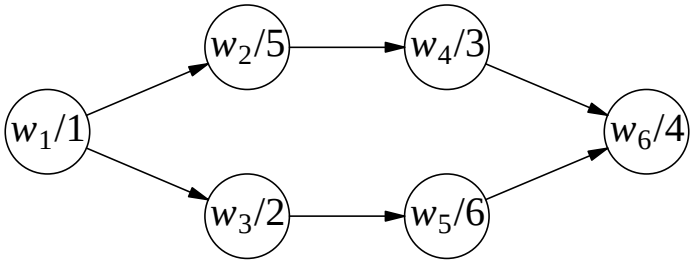
- $E_1 = p[1 + E_1] + q$. This gives $E_1 = 1/(1 - p) = 1/q > 1$.
- $E_2 = p[2 + E_2] + qp[2 + E_1] + 2q^2$.

  This gives $E_2 = (2p + 2qp + p + 2q^2)/q = (2 + p)/q > 2$.

**EXERCISE**

1. Show that $E_3 = (3 + 3p)/q$ and $E_4 = (4 + 6p)/q$.
2. Show that $E_n = [n + p.\, n(n - 1)/2]/q > n$. How does this justify $(t_{i+1} - t_i)^2$ as *MDM* if we assume that there is only one critical path? If there are multiple critical paths, then should we multiply $(t_{i+1} - t_i)^2$ by the *parallelism-degree of critical jobs* within the milestone (give examples to illustrates your argument)?
3. Suppose we break up the inspection period $n$ into two parts $n_1$ and $n_2$, where $n = n_1 + n_2$. Compute $E_n - (E_{n_1} + E_{n_2})$ and find the best choice of $n_1$ and $n_2$.
4. Let $E'_n$ = Average duration for error-free completion of $w$ if we cannot detect from the inspection after completion of $w_n$ the $w_i$ where the first error occurred (i.e., we must rework each of $w_1$, $w_2$, $\cdots$, $w_n$). Show the equation for $E'_n$ and that $E'_n = n/(q^n)$.
5. Show that $E'_n > 2.E_n$ for $n > 1$.
6. Argue that even in this new model, breaking an inspection period $n$ into two parts $n_1$ and $n_2$, where $n = n_1 + n_2$, is better, i.e., $E'_{n_1} + E'_{n_2} < E'_n$. What is the optimal choice of $n_1$ and $n_2$?

7. If we modify the $R(w_j)$ for the AON below by the expected values of $R(w_j)$ based on $p = 0.1$, then show the critical paths and the best possible schedule based on MDM.




8. The 100% rule in a WBS applies to costs of (and also to total man-hours required for) tasks associated with the nodes of the WBS. Why is it that the 100% rule does not apply, in general, to the time required to complete the tasks associated with the nodes of WBS. Is there is a special case when the 100% rules does apply with respect to the time?

# MILESTONES AND CRITICAL PATHS

**Necessary Condition:**

- Time $t$ is a milestone (in some schedule) only if $t$ is the completion-time of a critical task.

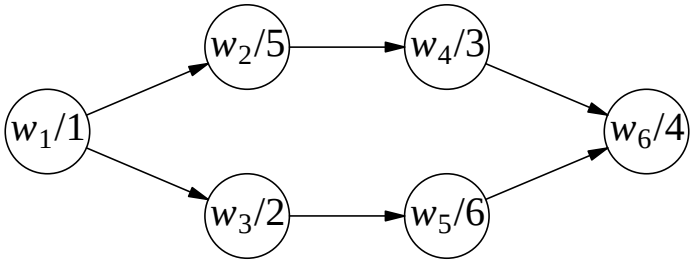
**Sufficient Condition:**

- If the critical path for the subset of tasks $EC(t) = \{w_i$: earliest completion time$(w_i) \leq t\}$ and a critical path for its complement set of tasks together form a critical path for all tasks, then there is a schedule with $t$ as a milestone.

**Question:**

•? Argue that the sufficient condition is also a necessary condition.

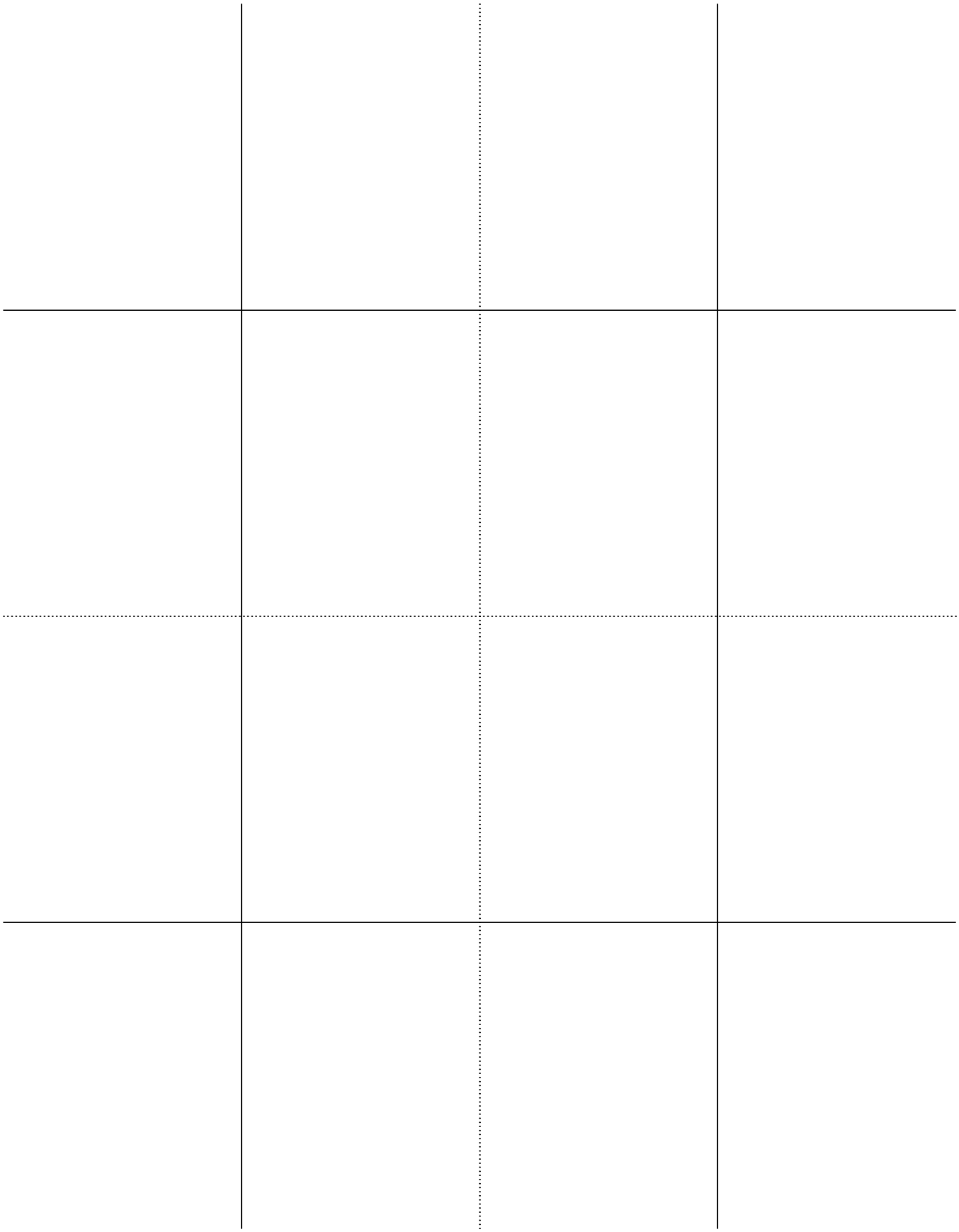
•? Verify this using the example AON below.

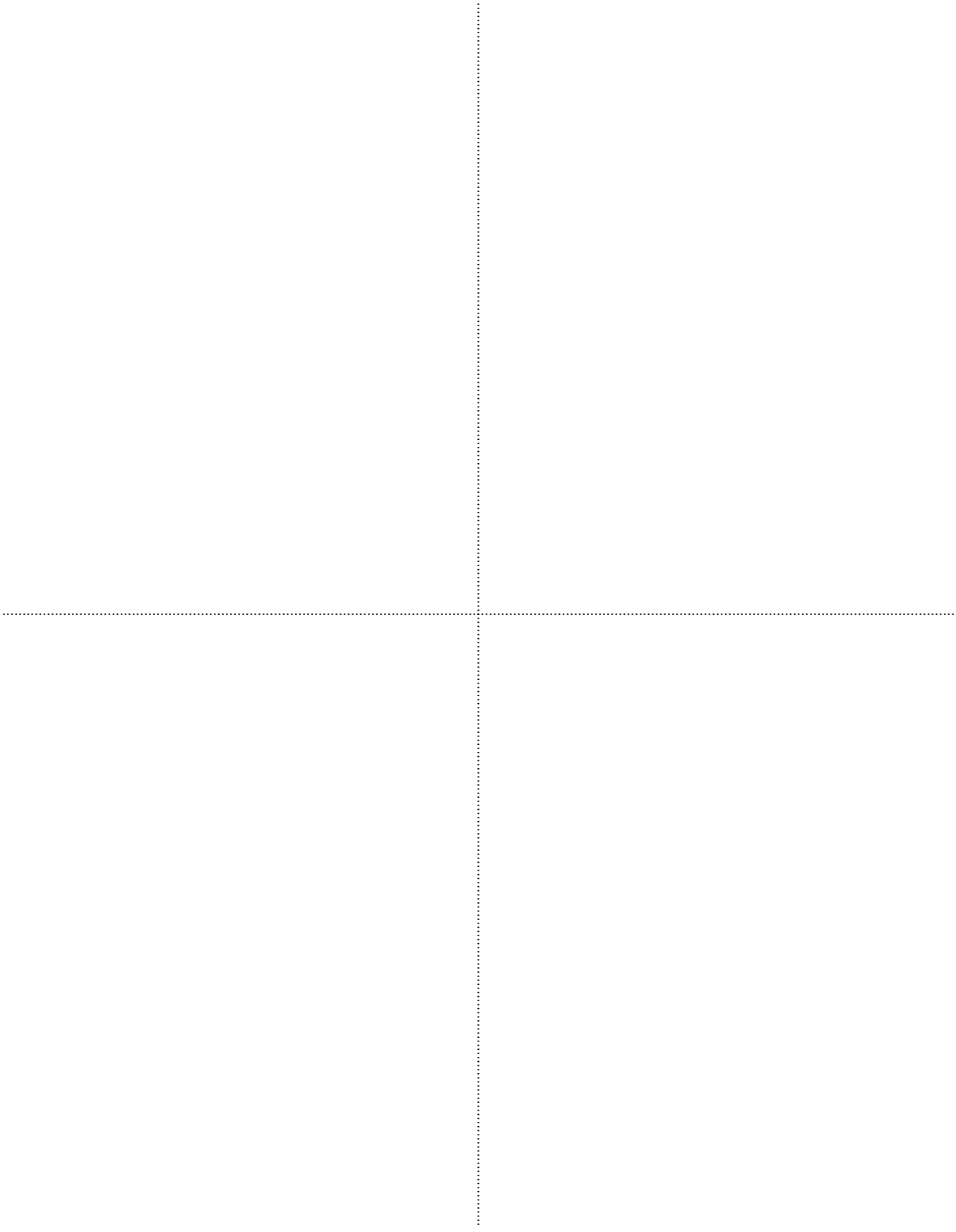



**Note:** If timepoints $t_1$ and $t_2$ satisfy the sufficient condition, it does not mean that both can be a milestone in the same schedule.

**Question:**

•? Modify $R(C)$ in the Gantt-Chart example to illustrate the point in the "note" above.





# PREPARING A PROJECT-IDEA DOCUMENT

**Choosing a Title:** Make it short and informative.

- It should help us focus/direct our thought and set expectations.
- Must be able to justify the presence of each word in the title, including arrangement of those words; avoid unnecessary words.

**Question:** Find one *key* word in each title; which word(s) would you remove if you had to?

| | |
|---|---|
| $T_1$: | Web Modeling Language (WebML): a modeling language for designing Web sites |
| $T_2$: | Science for Indigenous Children. |
| $T_3$: | Indigenous Science for Children. |
| $T_4$: | Science for Children. |
| $T_5$: | EasyTutor: Free, Unlimited, Online Tutoring Access |

**Novelty of Project #5:** Online.

- Makes the service easily and widely available, eliminating the geographic separation between students and tutors.
- Expands tutor availability; each student can be served by many tutors and each tutor can serve many students.
- Uses of tutor's expertise and time effectively; solutions to student-questions can be saved for online search/access and refined.

**Need – the Problem and the Project's Impact:**

- Serious lack of qualified tutors
- It benefits students from low-income/under-educated families (where parents can't provide tutoring).

**Question:** Make a project-idea document.

# A SAMPLE PROJECT-IDEA DOCUMENT

(from American Express Community Project Competition, 2009)

**The title:** Easy Tutor - Free, Online, Unlimited Access.

**The project:**

A website that connects students from around the nation to quality volunteer tutors through a live chat program. Students can chat individually with tutors as well as draw on virtual whiteboards and even use microphones to orally communicate back and forth. All tutoring will be free and available as long as tutors are online to take tutoring requests from students. The website should also offer an online community for students to interact on many academic topics and share knowledge and support.

**The Problem:**

Many students struggle through studying and homework and often enter the classroom not understanding the content previously taught. Hiring private tutors is often very expensive and simply improbable for many students. The internet is easily accessible from home, school, a local library, or community center so offering free tutoring through a website can make it available to anyone needing help. Tutors will provide support in any subject, allowing the student user to gain a proper understanding.

**The impact:**

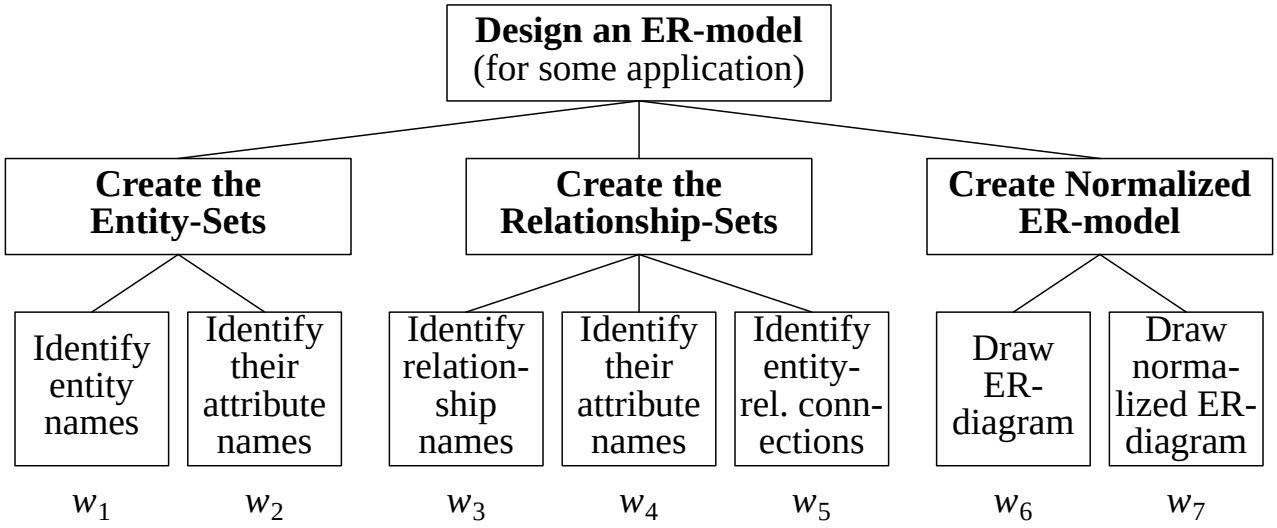
The project will have a widespread impact on schools and individual students around the world. Many students will now know that they have an alternative resource to look to when they feel that all hope is lost in school. Furthermore, the ability to accept volunteers as tutors will allow many who share a passion for teaching to do so in the comfort of their own homes through flexible scheduling.

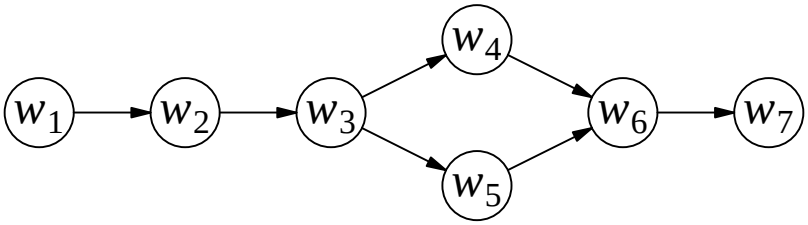
**The motivation:**

Having recently graduated from high school, I am aware of the need for supportive educational opportunities. I saw many of my peers struggling with no way out. They would come back unprepared to test, write an essay, or present a project. Had this project been in place, they would be able to go home and request the unlimited support of a tutor free of charge to make sure that they walk in the next day prepared and ready for anything. With this project, every single student would have a resource.

**EXERCISE** (Contd.)

3. Shown below is a WBS for designing an ER-model of a database (for some application) and an AON for it. Identify the problems with the AON and show a corrected one. (There are some problems with the WBS itself; see the next exercise.)




4. What is wrong with the WBS above?
   (a) Identify missing work-units if any and rename the work-units if needed; keep the nodes in bold letters unchanged.
   (b) Show the AON for the modified WBS. (If needed, reorganize the WBS tree to make it proper with respect to path-closed property.)