SHORTEST PATHS BY DIJKSTRA'S AND FLOYD'S ALGORITHM

Dijkstra's Algorithm:

- Finds shortest path from a given startNode to all other nodes reachable from it in a digraph.
- Assumes that each link cost $c(x, y) \geq 0$.
- Complexity: $O(N^2)$, N = #(nodes in the digraph)

Floyd's Algorithm:

- Finds a shortest-path for all node-pairs (x, y).
- We can have one or more links of negative cost, $c(x, y) < 0$, but no cycle of negative cost. (Assume that $c(x_i, x_i) = 0$ for each node x_i, which is the same as not having the links (x_i, x_i).)
- Complexity: $O(N^3)$, where N = #(nodes in digraph).

Can Dijkstra Be Used in Place of Floyd:

- If all $c(x, y) > 0$, then we can apply Dijkstra's algorithm for each x as a start-node. This takes $N . O(N^2) = O(N^3)$ time.

What Makes Floyd's Algorithm So Attractive:

- The number of acyclic paths from x_i to x_j in a complete digraph on N nodes can be $O((N-2)!)$.
- Floyd's algorithm computes the length of only $N+1$ paths from x_i to x_j, for each node-pair (x_i, x_j).

  Dijkstra's algorithm computes length of at most $N-1$ paths from the startNode to a node y to obtain a shortest path to y.

WEIGHTED DIGRAPH AND SHORTEST PATHS

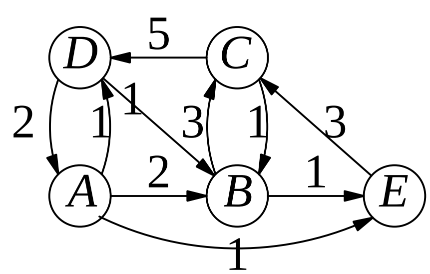
Weighted Digraph:

Each link (x, y) has a *length* or *cost* $c(x, y)$, which may or may not be positive; also, we may have $c(x, y) \neq c(y, x)$.

Path Length or Cost:

For a path $\pi = \langle x_1, x_2, \cdots, x_n \rangle$, its length is $|\pi| = c(x_1, x_2) + c(x_2, x_3) + \cdots + c(x_{n-1}, x_n)$.

A weighted digraph $\vec{G}$:



$|\langle A, B, E \rangle|$
$= 2+1 = 3$

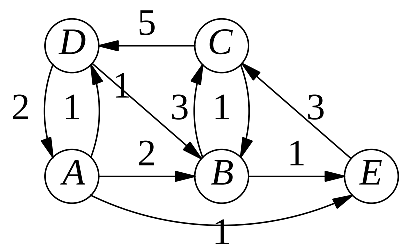
Shortest Path:

- $\pi_m(x, y)$ = an xy-path which has the smallest length among all xy-paths; if there is no xy-path, then we let $|\pi_m(x, y)| = \infty$.
- If an xy-path contains a node which is on a cycle of length < 0, then $\pi_m(x, y)$ is not well-defined (becomes $-\infty$).

  There is no shortest AD-path in $\vec{G}$ above if we add the link (C, E) with $c(C, E) = -4$; it gives a negative cycle at nodes on some AD-paths.
- Henceforth, assume that all links (x, y) have $x \neq y$.
- If $\vec{G}$ is acyclic, there are only a finite number of xy-paths for any x and y, and hence we always have a shortest xy-path if there is at least one xy-path.

AN EXAMPLE OF SHORTEST PATHS

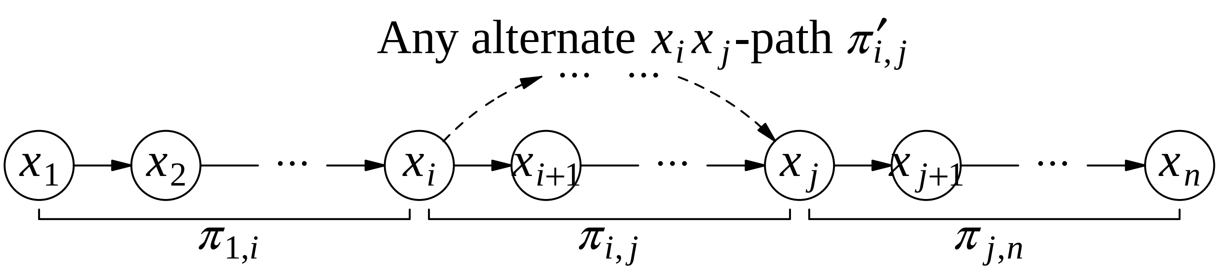
A weighted digraph $\vec{G}$:



	$y = A$	$y = B$	$y = C$	$y = D$	$y = E$
$x = A$	$\langle A, A\rangle$ 0	$\langle A, B\rangle$ 2	$\langle A, E, C\rangle$ 4=1+3	$\langle A, D\rangle$ 1	$\langle A, E\rangle$ 1
$x = B$	$\langle B, C, D, A\rangle$ 10=3+5+2	$\langle B, B\rangle$ 0	$\langle B, C\rangle$ 3	$\langle B, C, D\rangle$ 8=3+5	$\langle B, E\rangle$ 1
$x = C$	$\langle C, D, A\rangle$ 7=5+2	$\langle C, B\rangle$ 1	$\langle C, C\rangle$ 0	$\langle C, D\rangle$ 5	$\langle C, B, E\rangle$ 2=1+1

Bellman's Optimality Principle:

- Each subpath of a shortest path is itself a shortest path. That is, if $\pi = \langle x_1, x_2, \cdots, x_n\rangle$ is a shortest path, then each subpath $\pi_{i,j} = \langle x_i, x_{i+1}, \cdots, x_j\rangle$ is a shortest $x_i x_j$-path for $1 \le i < j \le n$.



$$\begin{aligned} |\pi_{1,i}| + |\pi_{i,j}| + |\pi_{j,n}| = |\pi| &\le |\text{the alternate } x_1 x_n\text{-path using } \pi'_{ij}| \\ &= |\pi_{1,i}| + |\pi'_{i,j}| + |\pi_{j,n}| \\ \text{i.e., } |\pi_{i,j}| &\le |\pi'_{i,j}|. \end{aligned}$$

Conclusion: Any method for finding a shortest xy-path is likely to find a shortest path between many other node-pairs.

Question:

•? What does the following shortest path $\pi_m(x_3, x_9)$ in some weighted digraph $\vec{G}$ say about some other shortest paths and their lengths?

$$x_3 \xrightarrow{15} x_5 \xrightarrow{8} x_2 \xrightarrow{18} x_4 \xrightarrow{7} x_9$$

•? Assume that $c(x, y) > 0$ for each link (x, y) and let $d(x, y) = |\pi_m(x, y)|$, with $d(x, x) = 0$ for all x. Which of the following are true?

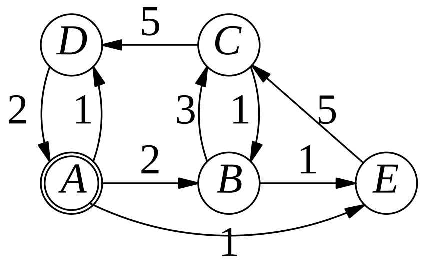
(i) $d(x, y) \geq 0$ for all x and y, and $= 0$ if and only if $x = y$.
(ii) $d(x, y) = d(y, x)$ for all x, y.
(iii) $d(x, z) \leq d(x, y) + d(y, z)$ for all x, y, and z.

If $c(x, y) = 0$ for one or more links (x, y), $x \neq y$, then which of the properties (i)-(iii) might cease to hold?

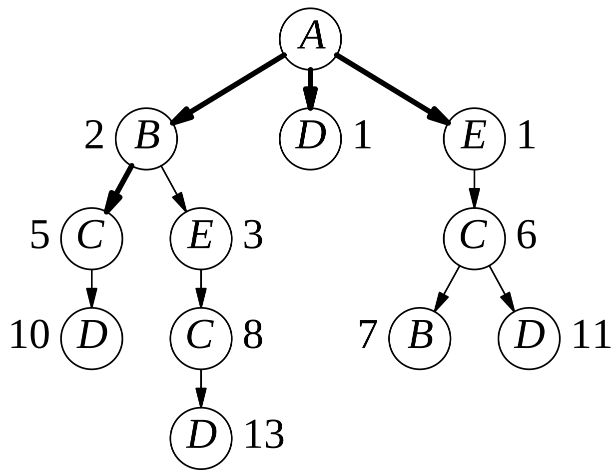
•? Under what condition can we delete a link (x_p, x_q) in $\vec{G}$ without affecting any $|\pi_m(x_i, x_j)|$, $1 \leq i, j \leq N$? Verify your solution using the digraph $\vec{G}$ on page 3.3. Can the deletions be performed in any order?

•? Assume that $\vec{G} = (V, \vec{E})$ contains no extraneous link (see the previous problem) and that each $c(x_i, x_j) > 0$. If you are given all $|\pi_m(x_i, x_j)|$, $1 \leq i \neq j \leq N$, how will you determine the links in $\vec{G}$ and the weights? Verify your solution using the digraph $\vec{G}$ on page 3.3. Does you solution work if one or more $c(x_i, x_j) \leq 0$ but there is no negative cost cycle?

TREE OF SHORTEST-PATHS IN DIGRAPHS



- A digraph $\vec{G}$ with each $c(x, y) \geq 0$.
- Start-node $s = A$.



- Bold links show the tree of shortest-paths to various nodes.

The tree of acyclic paths from A; shown next to each node is the length of the path from root = A.

Some Important Observations:

- Any subpath of a shortest path is a shortest path.
- The shortest paths from a startNode to other nodes can be chosen so that they form a tree.

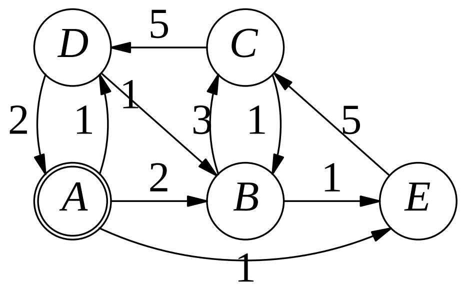
Question:

•? What are some minimum changes to the link-costs that will make $\langle A, B, E, C\rangle$ the shortest AC-path?

•? Show the new tree of acyclic paths and the shortest paths from startNode = A after adding the link (D, B) with cost 1.

•? Also show the tree of acyclic paths and the shortest paths from the startNode = D.

ILLUSTRATION OF DIJKSTRA'S SHORTEST-PATH ALGORITHM



- $d(x)$ = length of best path known to x from the startNode = A.
- A node is closed if shortest path to it known.
- A node is OPEN if a path to it known, but no shortest path is known.

"?,?" indicates unknown values, "⋯" indicates no changes, and "–" indicates path-length not computed (would not have changed any way).

Open Nodes	Node Closed	Links processed	$d(x)$ and parent(x) = node previous to x				
			A	B	C	D	E
$\{A\}$	$\emptyset$		0, A	?, ?	?, ?	?, ?	?, ?
$\{B\}$	A	(A, B)		2, A			
$\{B, D\}$		(A, D)				1, A	
$\{B, D, E\}$	(A, E)						1, A
$\{B, E\}$	D	(D, A)	–				
		(D, B)		⋯			
$\{B, C\}$	E	(E, C)			6, E		
$\{C\}$	B	(B, C)			5, B		
		(B, E)					–
$\emptyset$	C	(C, B)		–			
		(C, D)				–	

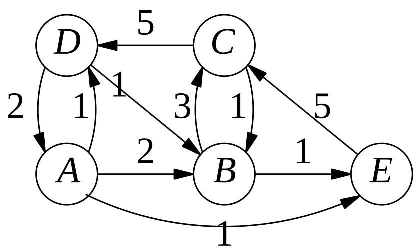
Question:

•? List all paths from A that are looked at (length computed) above.

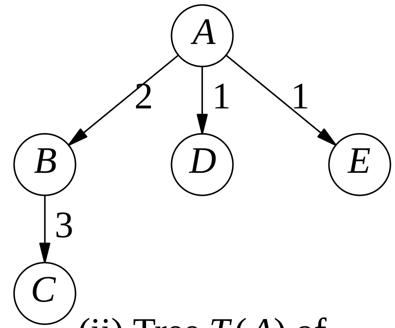
•? When do we look at a link (x, y)? How many times do we look at a link (x, y)?

•? What might happen if some $c(x, y) < 0$?

MORE ON BUILDING SHORTEST-PATHS



(i) A slightly modified form of $\vec{G}$ page 3.4; $c(E, C) = 5$.



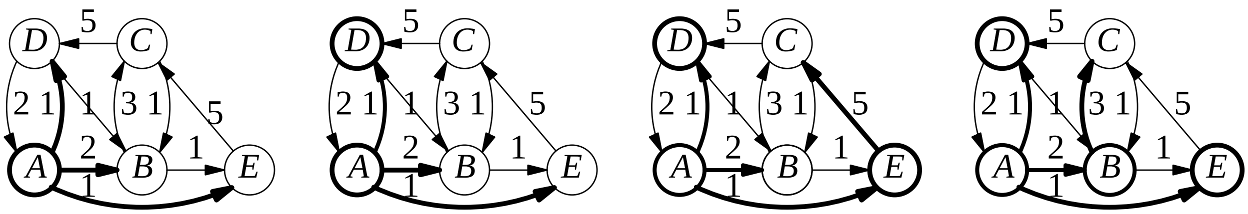
(ii) Tree $T(A)$ of shortest paths from A.

Method

- Exploits input-property "each $c(x, y) \geq 0$" and the output-property "tree-structure of shortest-paths from start-node s".
- Maintains a tree of currently best known paths $\pi(s, x)$ from s for various x; $d(x) = |\pi(s, x)|$.
- Extends $\pi(s, x)$ by adding links (x, y) from x *only* if $\pi(s, x) = \pi_m(s, x)$, i.e., x = a *terminal* node and $d(x)$ is minimum among terminal nodes. This step is called *closing* of x.

Successive States of the Tree of Current $\pi(s, x)$'s:

- The tree links are shown in bold.
- The closing of B gives the final tree $T(A)$.



(i) Closing $s = A$. (ii) Closing D. (iii) Closing E. (iv) Closing B; parent(C) changed.

DIJKSTRA'S ALGORITHM

Terminology (OPEN$\cap$CLOSED = $\emptyset$):

$\pi_m(s, x)$ = A shortest length sx-path (from s to x).
CLOSED = $\{x$: $\pi_m(s, x)$ is known$\}$.
OPEN = $\{x$: some $\pi(s, x)$ is known but $x \notin$ CLOSED$\}$.

Algorithm DIJKSTRA (shortest paths from s):

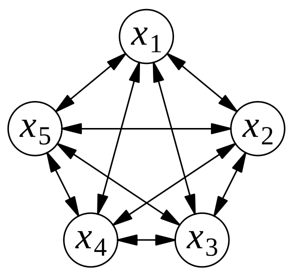
Input: AdjList(x) for each node x in $\vec{G}$, each $c(x, y) \geq 0$, a start-node s, and possibly a goal-node g.
Output: A shortest sg-path (or sx-path for each x reachable from s).

1. [Initialize.] $d(s) = 0$, mark s OPEN, parent(s) = s (or NULL), and all other nodes are unmarked.
2. [Choose a new closing-node.] If (no OPEN nodes), then there is no sg-path and stop. Otherwise, choose an OPEN node x with the smallest $d(\cdot)$, with preference for $x = g$. If $x = g$ or all but one node are closed, then stop.
3. [Close x and Expand $\pi_m(s, x)$.] Mark x CLOSED and for (each $y \in$ adjList(x) and y not marked CLOSED) do:
   if (y not marked OPEN or $d(x)+c(x, y) < d(y)$) then let parent(y) = x, $d(y) = d(x) + c(x, y)$, and mark y OPEN.
4. Go to step (2).

Complexity: $O(N^2)$.

- A node x is marked CLOSED at most once and hence a link (x, y) is processed at most once.
- Each iteration of steps (2) and (3) takes $O(N)$ time.

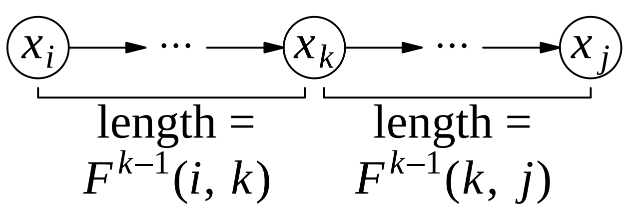
FLOYD'S METHOD FOR SHORTEST-PATHS FOR ALL NODE-PAIRS



- If $(x_i, x_j) \notin \vec{G}$, then let $c(x_i, x_j) = \infty$ (i.e., a large number L, say, $L = 1 + \sum|c(x_p, x_q)|$ (summed over all links in $\vec{G}$)
- Assume momentarily that each $c(x_i, x_j) \geq 0$ and each $c(x_i, x_i) = 0$.

Floyd's Equations:

- Let $F^k(i, j)$ = The shortest length of an $x_i x_j$-path which uses zero or more intermediate node from $\{x_1, x_2, \cdots, x_k\}$.
- For all $1 \leq i, j \leq N$ and $k \geq 1$,
  (1) $F^0(i, j) = c(x_i, x_j)$ if $i \neq j$ and $F^0(i, i) = 0$ for each i.
  (2) $F^k(i, j) = \min \begin{cases} F^{k-1}(i, j) & \text{(if } x_k \text{ is not used)} \\ F^{k-1}(i, k) + F^{k-1}(k, j) & \text{(if } x_k \text{ is used)} \end{cases}$
  (3) $F^N(i, j) = |\pi_m(x_i, x_j)|$, where N = #(nodes)



The case when x_k is used for $F^k(i, j)$.

Remarks:

- The equations (1)-(3) hold even if one or more $c(x_i, x_j) < 0$ as long as there is no cycle whose total cost is < 0. A negative cost cycle is detected if $F^k(i, i) < 0$ for some k and i.
- $F^k(i, j) \leq F^{k-1}(i, j)$, i.e., $F^k(i, j)$ gradually decreases to the final value $|\pi_m(x_i, x_j)|$ as k increases from 0 to N.

FLOYD'S SHORTEST PATH ALGORITHM FOR ALL NODE-PAIRS

Observation:

- For $k \geq 1$, $F^k(i, k) = F^{k-1}(i, k)$ and $F^k(k, j) = F^{k-1}(k, j)$ if no negative cycle is detected at the iteration for $(k-1)$.

$$\begin{aligned} F^k(i, k) &= \min \begin{cases} F^{k-1}(i, k), \\ F^{k-1}(i, k) + F^{k-1}(k, k) \end{cases} \\ &= F^{k-1}(i, k) \end{aligned}$$

Similarly, $F^k(k, j) = F^{k-1}(k, j)$.

Algorithm FLOYD:

Input: The link-costs $c(x_i, x_j)$ of a digraph on N nodes; $c(x_i, x_i) = 0$ for each $1 \leq i \leq N$.

Output: The costs $F[i, j]$ of an optimal $x_i x_j$-path for all $1 \leq i, j \leq N$ if there is no negative cycle.

1. [Initialize.] For $(1 \leq i, j \leq N)$, let $F[i, j] = c(x_i, x_j)$.
2. For $(k = 1, 2, \cdots, N)$ do the following:
   - For $(1 \leq i, j \leq N$ and $k \neq i, j)$, let $F[i, j] = \min\{F[i, j], F[i, k] + F[k, j]\}$.
   - If (some $F[i, i] < 0$) stop.

Complexity: = $\Theta(N^3)$:

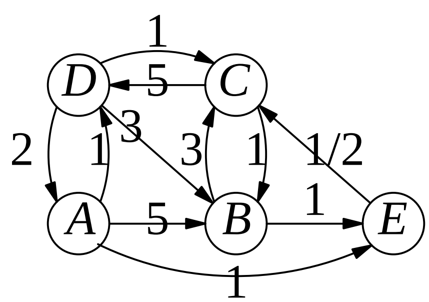
- Step (1) takes $\Theta(N^2)$ time.
- Each iteration for k in step (2) takes $\Theta(N^2)$ time.

KEEPING TRACK OF SHORTEST PATHS

Path $\pi^k_{i,j}$: An $x_i x_j$-path corresponding to $F^k(i, j)$, i.e., has length $F^k(i, j)$ and uses only the nodes $\{x_1, x_2, \cdots, x_k\}$ as possible intermediate nodes.

- Let $Next^k(i, j)$ = the node next to x_i on $\pi^k_{i,j}$.
- Compute $Next^k(i, j)$ along with $F^k(i, j)$ as follows.
  (1) $Next^0(i, j) = j$
  (2) $Next^k(i, j) = \begin{cases} Next^{k-1}(i, j), & \text{if } F^k(i, j) = F^{k-1}(i, j) \\ Next^{k-1}(i, k), & \text{if } F^k(i, j) < F^{k-1}(i, j) \end{cases}$
- The final $x_i x_j$-path is given by $\langle x_i, x_{i_1}, x_{i_2}, \cdots, x_j \rangle$, where $i_1 = Next^N(i, j)$, $i_2 = Next^N(i_1, j)$, and so on.

Example. Consider a slightly different digraph shown below.



$F^3(A, B) = 5$, $Next^3(A, B) = B$
$F^3(D, B) = 2$, $Next^3(D, B) = C$

$F^4(A, B) = 3$, $\pi^4_{A,B} = \langle A, D, C, B \rangle$, $Next^4(A, B) = D$.
$F^4(D, B) = 2$, $\pi^4_{D,B} = \langle D, C, B \rangle$, $Next^4(D, B) = C$.
$F^4(C, B) = 1$, $\pi^4_{C,B} = \langle C, B \rangle$, $Next^4(C, B) = B$.

Question:

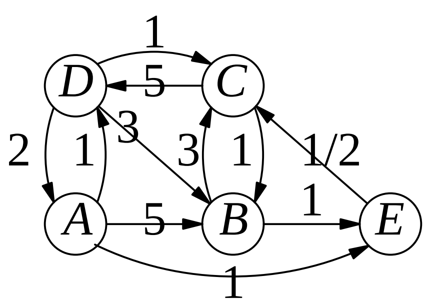
•? Show the successive values of $Next^5(\cdot, \cdot)$ in relation to $F^5(A, B)$ and $\pi^5_{A,B}$.

ALTERNATE METHOD FOR KEEPING TRACK OF SHORTEST PATHS

- Let $Best^k(i, j) = \min\{p: F^p(i, j) = F^k(i, j)\}$.
- Compute $Best^k(i, j)$ along with $F^k(i, j)$ as follows:

  (1) $Best^0(i, j) = 0$ (initialization).
  (2) $Best^k(i, j) = k$ if $F^k(i, j) < F^{k-1}(i, j)$ for $k \geq 1$; otherwise, $Best^k(i, j) = Best^{k-1}(i, j)$.

Example. Consider the digraph shown below.



$F^3(A, B) = 5$, $Best^3(A, B) = 0$
$F^3(D, B) = 2$, $Best^3(D, B) = 3$

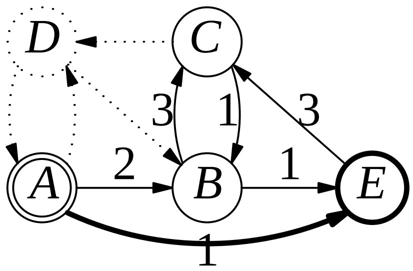
$F^4(A, B) = 3$, $\pi^4_{A,B} = \langle A, D, C, B \rangle$, $Best^4(A, B) = 4$.
$F^3(A, D) = 1$, $\pi^3_{A,D} = \langle A, D \rangle$, $Best^3(A, D) = 0$.
$F^3(D, B) = 2$, $\pi^3_{D,B} = \langle D, C, B \rangle$, $Best^3(D, B) = 3$.
$F^2(D, C) = 1$, $\pi^2_{D,C} = \langle D, C \rangle$, $Best^2(D, C) = 0$.
$F^2(C, B) = 1$, $\pi^2_{C,B} = \langle C, B \rangle$, $Best^2(C, B) = 0$.

Question:

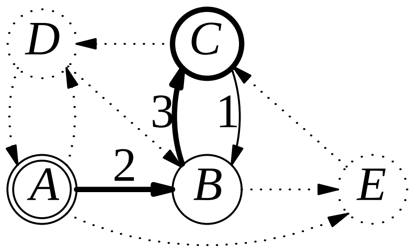
•? Show the successive values of $Best^k(\cdot, \cdot)$ in relation to $F^5(A, B)$ and $\pi^5_{A,B}$.

•? Give a pseudocode for constructing a shortest $x_i x_j$-path $\pi_m(x_i, x_j)$ from $Best^N(i, j)$'s.

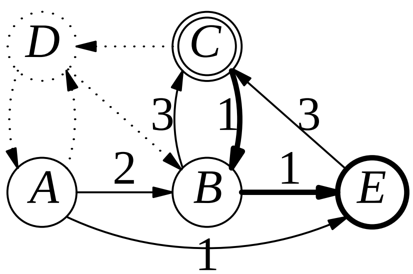
THE NEW PATH $\langle A, B, C, B, E\rangle$ EXAMINED IN COMPUTING $F^3(A, E)$



The part of $\vec{G}$ used for computing $F^3(A, E) = 1$, with nodes A and E specially marked; $\pi^2_{A,E} = \langle A, E\rangle$.

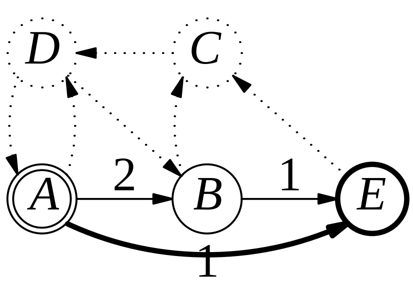


The part of $\vec{G}$ used for computing $F^2(A, C) = 5$, with nodes A and C specially marked; $\pi^2_{A,C} = \langle A, B, C\rangle$.



The part of $\vec{G}$ used for computing $F^2(C, E) = 2$, with nodes C and E specially marked; $\pi^2_{C,E} = \langle C, B, E\rangle$.

$$F^3(A, E) = 1 = \min\begin{cases} F^2(A, E) = 1, \\ F^2(A, C) + F^2(C, E) = 5 + 2 = 7 \end{cases}$$



The part of $\vec{G}$ used for computing $F^2(A, E) = 1$, with nodes A and E specially marked; $\pi^2_{A,E} = \langle A, E\rangle$.

Question: Do we examine the path $\langle C, D, A, E\rangle$ in $\vec{G}$ shown on previous page – explain your answer.

MOST ACYCLIC PATHS ARE NOT EXAMINED IN FLOYD'S ALGORITHM

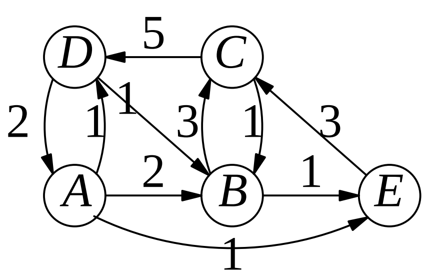
One Additional $x_i x_j$-path is Examined per Iteration:

- In computing $F^{k-1}(i, k)+F^{k-1}(k, j)$ we implicitly consider the path $\pi_{i,k}^{k-1}.\pi_{k,j}^{k-1}$ for $k \geq 1$ and $k \neq i, j$.

Total Number of $x_i x_j$-Paths Considered:

- At most $N + 1$, which may include some cyclic paths.
- Most of $\Theta((N-2)!)$ acyclic $x_i x_j$-paths are not examined.

Total Number of Paths Considered: $\Theta(N^3)$.

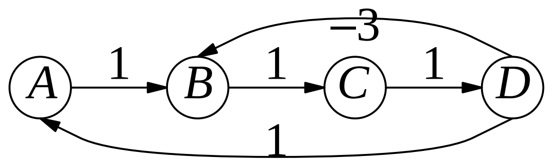


- All links not shown have cost 20.
- $x_1 = A$, $x_2 = B$, $\cdots$, $x_5 = E$.

Path $\langle A, B, C, E\rangle$ Not Examined:

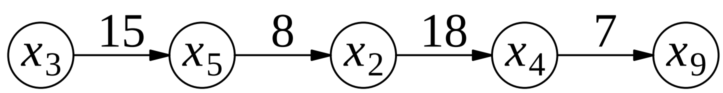
- If this path were examined in computing $F^k(A, E)$, i.e., $F^k(1, 5)$, then $k = 3$ (why?). However, the new path examined in computing $F^3(A, E)$ is $\pi_{A,C}^2.\pi_{C,E}^2 = \langle A, B, C, B, E\rangle$.

Question: What is the smallest k such that $F^k(i, i) < 0$ for some i for the digraph below? What is that i?

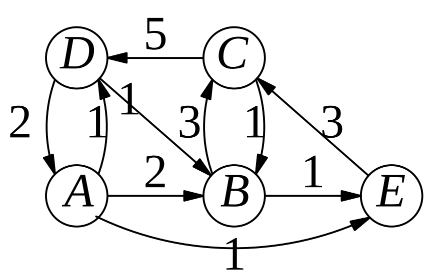


EXERCISE

1. Suppose the following path is a shortest x_3x_9-path in some digraph G with non-negative link-costs. For each shortest sub-path $\pi_m(x_i, x_j)$ of $\pi_m(x_3, x_9)$, indicate the smallest k (which may depend on i and j) such that we will *definitely* have $F^k(i, j) = |\pi_m(x_i, x_j)|$ no matter what the costs of other links in G are. Give an example to show that we may have $F^n(i, j) = |\pi_m(x_i, x_j)|$ for some $n < k$ in some cases, though this may not be guaranteed for all costs.



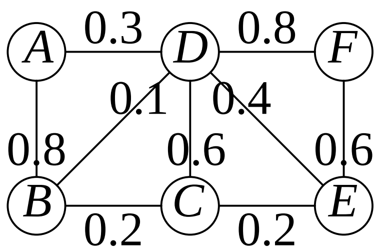
2. Show the matrices $F^4[i, j]$ and $Next^4[i, j]$, $1 \le i, j \le 5$, for the digraph below. Also, show all AE-paths and EA-paths whose lengths are computed by Floyd's algorithm.



- All links not shown have cost 20.
- $x_1 = A$, $x_2 = B$, $\cdots$, $x_5 = E$.

3. Suppose $0 \le c(x, y) \le 1$ for each link (x, y) and $a \wedge b = \min(a, b)$. We define the cost of a path $\pi(x_1, x_n) = \langle x_1, x_2, \cdots, x_n \rangle$ by $C_{\min}(\pi) = c(x_1, x_2) \wedge c(x_2, x_3) \wedge \cdots \wedge c(x_{n-1}, x_n)$ and let $M_{\min}(x, y) = \max\{C_{\min}(\pi)$: for all paths $\pi = \pi(x, y)\}$. If we think of $c(x, y)$ as the traffic flow from x to y, then $C_{\min}(\pi)$ is the traffic flow along the path π.

   (a) Show $M_{\min}(A, B)$, $M_{\min}(B, C)$, and $M_{\min}(A, C)$ and a corresponding path for each case for the digraph below.



- $c(x, y) = c(y, x)$ for all (x, y).
- $c(x, y) = 0$ if (x, y) is not present.
- $c(x, x) = 1$ for all x.

(b) Is it true that $M_{\min}(x, z) \geq \min\{M_{\min}(x, y), M_{\min}(y, z)\}$ for all x, y, and z? Explain.

(c) Which of the equations (i)-(iii) in Problem 2 on page 3.5 are still true for $d(x, y) = M_{\min}(x, y)$ and why?

(d) Write recursive equations (similar to Floyd's equations) for computing $M_{\min}(x, y)$.

(e) Is it true that we need to consider only the acyclic-paths in computing $M_{\min}(x, y)$ (why)?

4. [Recursive approach to computing all shortest-path lengths $d(i, j) = |\pi_m(x_i, x_j)|$.] Assume that you have computed the shortest-path lengths $d_N(i, j)$, $1 \leq i, j \leq N$, in the digraph G_N on the first N nodes $\{x_1, x_2, \cdots, x_N\}$. Show how to compute $d_{N+1}(i,j)$, $1 \leq i, j \leq N+1$, in the digraph G_{N+1} using the results for G_N. Verify your results using the digraph in Problem 2 (with $A = x_1$, $B = x_2$, etc.) and show the 4×4 matrix for $d_4(i, j)$'s and the 5×5 matrix of $d_5(i, j)$'s. Show the additional term in $T(N+1) = T(N) + \cdots$ using the appropriate notation $O(\cdot)$, or $\Theta(\cdot)$, or $\Omega(\cdot)$, where $T(N+1)$ is the time required to compute $d_{N+1}(i, j)$'s; also, express $T(N)$ in the form $O(\cdot)$, $\Omega(\cdot)$, or $\Theta(\cdot)$, as appropriate.

5. Let $count^k(i, j)$ = #(acyclic $x_i x_j$-paths corresponding to $F^k(i, j)$, i.e., with length $F^k(i, j)$ and $\{x_1, x_2, \cdots, x_k\}$ as possible intermediate nodes). Argue that the equations for computing $count^k(i, j)$ below are correct. For each k, $1 \leq k \leq 6$, show the matrix of $count^k(i, j)$ for the digraph with $N = 6$ nodes and the

costs $c(x_i, x_j) = |i - j|$ for $1 \le i \ne j \le N$; show only the non-zero counts for improved readability. List the paths corresponding to $count^4(2, 6)$.

$$count^k(i, j) = \begin{cases} count^{k-1}(i, k) \times count^{k-1}(k, j), \\ \quad \text{if } F^{k-1}(i, j) > F^{k-1}(i, k) + F^{k-1}(k, j) \\ count^{k-1}(i, j), \\ \quad \text{if } F^{k-1}(i, j) < F^{k-1}(i, k) + F^{k-1}(k, j) \\ count^{k-1}(i, k) \times count^{k-1}(k, j) + count^{k-1}(i, j), \\ \quad \text{if } F^{k-1}(i, j) = F^{k-1}(i, k) + F^{k-1}(k, j) \end{cases}$$

6. Let $\overline{count}^k(i, j)$ = #(acyclic $x_i x_j$ paths using the nodes $\{x_1, x_2, \cdots, x_k\}$ as possible intermediate nodes and which have length > $F^k(i, j)$). Use the formula for the total number of acyclic paths among which $F^k(i, j)$ is the $x_i x_j$ shortest-path length and $count^k(i, j)$ to obtain a formula for $\overline{count}^k(i, j)$. Why is it not possible to obtain formulas for $\overline{count}^k(i, j)$ in a way similar to those for $count^k(i, j)$?

7. Let G be an *acyclic* digraph. Give suitable recursive equations for computing $numPaths_{ij}$ = #($x_i x_j$-paths using one or more links in G). Note that $numPaths_{ii} = 0$ for all i. If G is not acyclic, what will go wrong with the equations? What will be the equations if we define $numPaths_{ij}^{k+1}$ = #($x_i x_j$-paths using exactly $k+1$ steps)? How about if $numPaths_{ij}^{k+1}$ = #($x_i x_j$-paths using at most $k+1$ steps)?

PROGRAMMING EXERCISE

1. Assume the input digraph is given in the form of adjacency lists, including the cost of the links. Shown below is the first few lines in the input file for the digraph on page 3.12.

   5 //numNodes; adjList items are adjNode(linkCost)
   0 (3): 1(5.0) 3(1.0) 4(1.0)
   1 (2): 2(3.0) 4(1.0)
   ...

   Write a function LengthAndNext **Floyd() which returns the array of shortest-path lengths $F^N[i, j]$ and $Next^N[i, j]$ based on the structure LengthAndNext (see below).

```
typedef struct {
 double pathLength;
 int nextNodeOnPath;
} LengthAndNext;
```

   Your output should show the 2-dimensional matrix; for example, for row 0 (node A) and column 2 (node C), the output should be "1.5: 4", where 1.5 gives the length of the shortest AC-path $\langle A, E, C\rangle$ and for the next node E after A on that path.

   Show the output for the input digraph as on page 3.12 in the lecture notes (http: //www,csc.lsu.edu/~kundu/dstr/notes.html). Show all distances upto one digit after the decimal point (as was in the input); use a separate function for output.

A VARIATION OF FLOYD'S METHOD

The New Equations:

- Let $Z^k(i, j)$ = The shortest length of an $x_i x_j$-path with *at most k* intermediate nodes (assume no negative cycle in $\vec{G}$).

  If $\vec{G}$ has $N = 5$, then the paths considered in the definition of $Z^1(4, 5)$ are the following; we may exclude the two paths marked '*' which have loops:

  $$\begin{array}{llll} \langle 2, 5\rangle & \langle 2, 1, 5\rangle & \langle 2, 4, 5\rangle & *\langle 2, 2, 5\rangle \\ & \langle 2, 3, 5\rangle & & *\langle 2, 5, 5\rangle \end{array}$$

  In contrast, the definition of $F^1(2, 5)$, we consider the only loop-free paths $\langle 2, 5\rangle$ and $\langle 2, 1, 5\rangle$.

- $Z^0(i, j) = c(x_i, x_j)$

  $$Z^k(i, j) = \min \begin{cases} Z^{k-1}(i, j) \\ Z^{k-1}(i, q) + c(x_q, x_j), \; 1 \le q \le N \end{cases}$$

$$\boxed{\begin{array}{l} (1)\ \ |\pi_m(i, j)| \le Z^k(i, j) \le Z^{k-1}(i, j) \le F^{k-1}(i, j) \le c(x_i, x_j) \\ (2)\ \ |\pi_m(i, j)| = Z^{N-2}(i, j) \text{ for } i \ne j \end{array}}$$

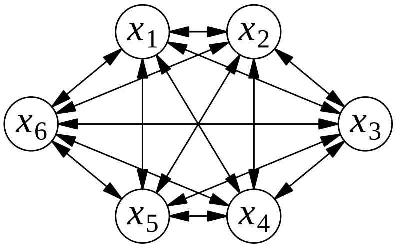
Comparison With Floyd:

- Complexity = $\theta(N^4)$, with $\Theta(N)$ work for each $Z^k(i, j)$.
- $F^k(i, j)$ converges to $|\pi_m(i, j)|$ slower than $Z^k(i, j)$ and yet takes less time for computation.
- This is also a D.P. method, with the same states as in Floyd's method, but with a different meaning of the states and a corresponding different relationship among them.
- Not all D.P. methods are equally good.

EXERCISE.

1. How many acyclic paths are considered in the definition of $Z^k(i, j)$ for $i \neq j$? How many paths (may not be acyclic) are considered or looked at in the equation for $Z^k(i, j)$, $i \neq j$?
2. Which of the paths $\langle A, C, B\rangle$ and $\langle A, C, B, D\rangle$ in $\vec{G}$ on page 3.3 are looked at by Floyd's and the Z^k-method (explain)?
3. Can we say $Z^k(i, j) = \min\{Z^{k-1}(i, j), c(x_i, x_q) + Z^{k-1}(q, j): 1 \leq q \leq N\}$? Will both forms of Z^k look at the same paths?

FINDING AN n-STEP SHORTEST-PATH



- $\vec{G}$ may have a negative-cost cycle.
- $S^n(i, j)$ = The shortest length of an n-step $x_i x_j$-path.
- We are not restricting to acyclic paths as verifi cation of acyclicity becomes expensive.

Example. Paths $\pi(1, 6)$ using 3 steps.

$\langle 1, 1, 1, 6\rangle$, $\langle 1, 1, 2, 6\rangle$, $\cdots$

The length of shortest path in this group is $c(x_{1,}\ x_2) + S^2(2, 6)$

$$\begin{cases} \langle 1, 2, 1, 6\rangle, \langle 1, 2, 2, 6\rangle, \\ \langle 1, 2, 3, 6\rangle, \langle 1, 2, 4, 6\rangle, \\ \langle 1, 2, 5, 6\rangle, \langle 1, 2, 6, 6\rangle \end{cases}$$

$\cdots\ \cdots$

$$S^3(1, N) = \min_{1 \le p \le N} \{c(x_1, x_p) + S^2(p, N)\}$$

General Case:

- $S^1(i, j) = c(x_i, x_j)$ for all $i \ne j$

  $S^n(i, j) = \min_{1 \le p \le N} \{c(x_i, x_p) + S^{n-1}(p, j)\}$, for $n \ge 2$

  Also, $S^n(i, j) = \min_{1 \le p \le N} \{S^{n-1}(i, p) + c(x_p, x_j)\}$, for $n \ge 2$

- One can also keep track of the paths along with the path-length computations.

Complexity: $\Theta(N^2 + (n-1). N. N^2) = \Theta(n. N^3)$ for n-step shortest paths for all (i, j)-pairs.

Question: What is #(n-step $x_i x_j$-paths)? What does it say about the above method for computing $S^n(i, j)$?

HISTOGRAM EQUALIZATION

Problem: Decompose the list of numbers $L = \langle n_1, n_2, \cdots, n_N \rangle$ into $k > 1$ groups of consecutive items (in short, a k-grouping) such that: each group-total is as close to the ideal value T/k, where $T = n_1 + n_2 + \cdots + n_N$. More precisely, if s_i = ith group-total, then minimize

$$E = \sum_{i=1}^{k} (s_i - T/k)^2 = \text{the sum of squared errors.}$$

Example. $L = \langle 3, 2, 1, 1, 2 \rangle$ and $k = 3$; $T/k = 3$.

The groups in a 3-grouping			Group totals
$\langle 3 \rangle$	$\langle 2 \rangle$	$\langle 1, 1, 2 \rangle$	3, 2, 4
$\langle 3 \rangle$	$\langle 2, 1 \rangle$	$\langle 1, 2 \rangle$	3, 3, 3 ← optimal solution
$\langle 3 \rangle$	$\langle 2, 1, 1 \rangle$	$\langle 2 \rangle$	3, 4, 2
$\langle 3, 2 \rangle$	$\langle 1 \rangle$	$\langle 1, 2 \rangle$	5, 1, 3
$\langle 3, 2 \rangle$	$\langle 1, 1 \rangle$	$\langle 2 \rangle$	5, 2, 2
$\langle 3, 2, 1 \rangle$	$\langle 1 \rangle$	$\langle 2 \rangle$	6, 1, 2

Question:

•? Show that #(k-groupings) = C^{N-1}_{k-1}; verify the formula for k = 1, 2, 3. (Hint: consider k-step paths in the digraph $\vec{G}_N$ considered in the next page.) How to generate them systematically and how long will it take?

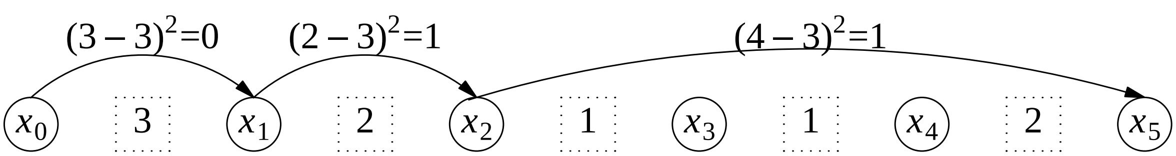
•? If $s_{i,j} = n_{i+1} + n_{i+2} + \cdots + n_j$, then show a table of all $s_{i,j}$ that are relevant in the optimal k-grouping problem? How many of them are there and what is the complexity for an efficient algorithm for computing them?

•? Give the pseudocode for finding an optimal k-grouping of N items using the $s_{i,j}$'s. Give its complexity.

k-STEP SHORTEST-PATH FORMULATION

Digraph $\vec{G}_N$:

- $V = \{x_0, x_1, \cdots, x_N\}$, $\vec{E} = \{(x_i, x_j)\colon 0 \le i < j \le N\}$, where the link (x_i, x_j) corresponds to the group $\{n_{i+1}, n_{i+2}, \cdots, n_j\}$.
- Each k-step $x_0 x_N$-path give a k-grouping and vice-versa.
- Let $c(x_i, x_j) = (s_{i,j} - T/k)^2$, where $s_{i,j} = n_{i+1} + n_{i+2} + \cdots + n_j$.
- A shortest k-step $x_0 x_N$-path gives an optimal k-grouping.



For $L = \langle 3, 2, 1, 1, 2\rangle$ and the 3-step $x_0 x_5$-path $\langle x_0, x_1, x_2, x_5\rangle$,
the associated 3-grouping is $\{\langle 3\rangle, \langle 2\rangle, \langle 1,1,2\rangle\}$, with cost 0+1+1 = 2.

Question

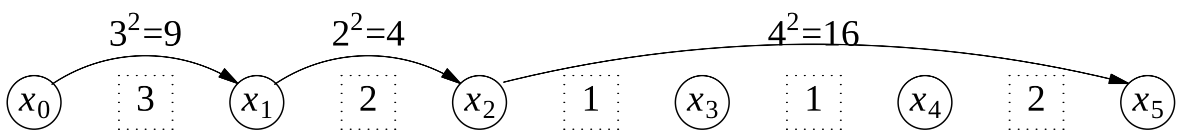
- ? Show the weighted digraph $\vec{G}_5$ for $L = \langle 3, 2, 1, 1\ , 2\rangle$ and the table of $S^n(0, j)$, $n \le j$ and $1 \le n \le 3$.
- ? Explain why the following is true for the digraph $\vec{G}_N$, where all links go from left-to-right: $S^k(0, j) = \min\ \{S^{k-1}(0, p) + c(x_p, x_j)\colon k - 1 \le p < j\}$.
- ? Verify your formula for $k = 3$ and $L = \langle 3, 2, 1, 1, 2\rangle$. Mark the items $S^n(0, j)$ that will be computed in the process.
- ? Give the exact number of additions involving one or more $c(x_i, x_j)$ in computing $S^k(0, N)$; show sufficient details.
- ? What is the (total) complexity of this method in terms of k and N? Why can we call this a D.P. method?

AN ALTERNATIVE LINK-COSTS FOR HISTOGRAM-EQUALIZATION

- Consider a k-steps $x_0 x_N$-path $\pi = \langle x_0,\ x_{i_1},\ x_{i_2},\ \cdots,\ x_{i_{k-1}},\ x_N \rangle$. Let $s_1 = s_{0,i_i}$, $s_2 = s_{i_1,i_2}$, $\cdots$, $s_k = s_{i_{k-1},N}$ and $a = T/k$. Then,

$$\begin{aligned}\mathrm{cost}(\pi) &= (s_1 - a)^2 + (s_2 - a)^2 + \cdots + (s_k - a)^2 \\ &= [s_1^2 + s_2^2 + \cdots + s_k^2] - 2a[s_1 + s_2 + \cdots + s_k] + ka^2 \\ &= [s_1^2 + s_2^2 + \cdots + s_k^2] - 2aT + ka^2\end{aligned}$$

- Minimizing cost(π) is the same as minimizing $s_1^2 + s_2^2 + \cdots + s_k^2$.
- This means we can replace the link-cost $(s_i - a)^2$ simply by s_i^2, which is independent of k and T/k.



Some of the new link-costs for $L = \langle 3, 2, 1, 1, 2 \rangle$; the new cost of path $\langle x_0,\ x_1,\ x_2,\ x_5 \rangle$ is 29 and the old cost = 2 = $29 - 2.3.9 + 3.3^2$.

The $S^n(0,\ j)$'s to be computed to obtain $S^3(0,\ N)$ for N = 5.

- Let $S[n,\ j]$ denote $S^n(0,\ j)$ for $n \le j$ ('–' means not computed).

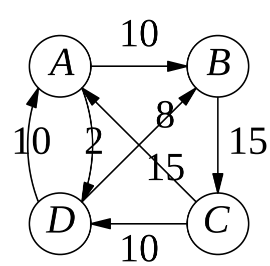
	x_0	x_1	x_2	x_3	x_4	x_5
$S[1, \cdot]$:	–	9	25	36	–	–
$S[2, \cdot]$:	–	–	13	18	25	–
$S[3, \cdot]$:	–	–	–	14	17	27

Computing Link-Costs: //not all are used in finding $S^k(0,\ N)$

```
for (i=0; i<numItems; i++)
    for (sum=0, j=i; j<numItems; j++) {
        sum += items[j];
        linkCosts[i][j+1] = sum*sum; }
```

EXERCISE

1. Consider the problem of creating $k = 3$ best possible grouping of the data-items $\langle 1, 2, 3, 1, 2 \rangle$, with $N = 5$. Consider the two different link-cost structures (with and without using the average $a = 9/3 = 3$) and for each of them show the relevant values of $S^n(i, j)$ in the table form for computing $S^3(0, 5)$. Also, for each of these values of $S^n(i, j)$, show the associated path in the same table (use the node names $x_0, x_1, \cdots, x_5$ to specify the paths).
2. For Problem 1 above, list all possible 3-step paths from x_0 to x_5 in the digraph $\vec{G}_5$ and list the paths whose length are not computed. For the general case $N \geq k$, what is the total number $S^n(i,j)$ values that we will compute to determine $S^k(0, N)$?
3. **Travelling salesman problem** (TAP): Given a digraph with link costs $c(x, y) \geq 0$, find a hamiltonian cycle (i.e., a cycle that goes through each node exactly once) which has the smallest cost. For the digraph below,



   (a) Why is it that we can drop the assumption $c(x, y) \geq 0$? (In other words, the condition $c(x, y) \geq 0$ does not make it easier or more difficult to solve TSP.)

   (b) Why can't we use the shortest-path algorithm to solve the TSP?

   (c) Why can't we use the k-step shortest-path algorithm to solve the TSP?

ASSIGNING LETTER-GRADES TO TEST-SCORES

Problem:

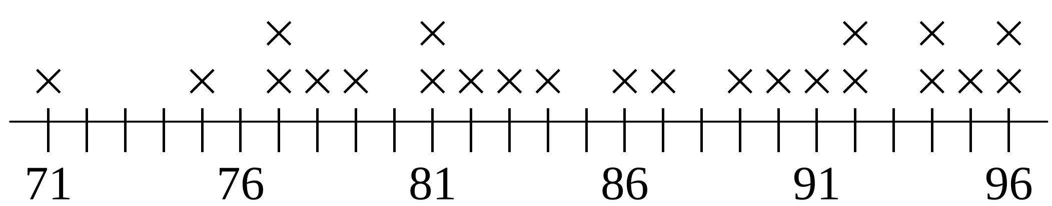
- We are given the distinct scores $0 \leq s_1 < s_2 < \cdots < s_m \leq 100$ in a class-test and the frequency n_i for each score s_i (n_i = number of students with score s_i). How do we determine the cut off points for the letter-grades A, B, C, D, and F?

Solution: By formulating it as a 5-Step shortest path problem.

- The scores for each letter-grade are viewed "equivalent" in some sense. Also, they are considered "significantly" different from the scores for the other letter-grades.

  This suggests the cost associated with a group $g_i = \{s_j, s_{j+1}, \cdots, s_k\}$ to be the variance $V(g_i)$.

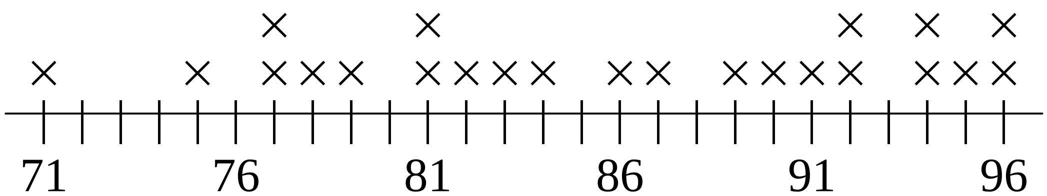
- Since the letter-grade "A" has higher GPA than that of "B", we should have $V(g_A) \leq V(g_B)$, and so on. This means we should minimize
  $$w_A V(g_A) + w_B V(g_B) + w_C V(g_C) + w_D V(g_D) + w_F V(g_F)$$
  with, say, $w_A = 4$, $w_B = 3$, $w_C = 2$, $w_D = 1$, and $w_F = 1$ (> 0).

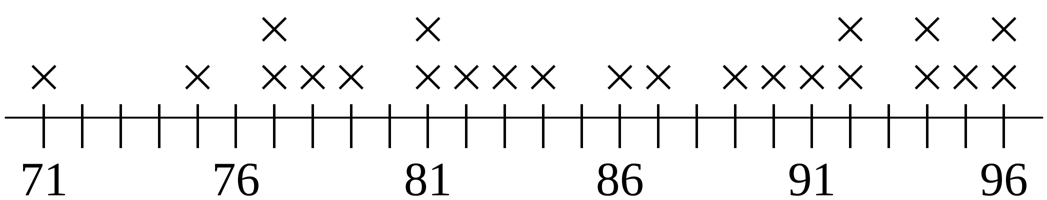


Scores of 23 students in a course (one ’×’ per student).

Question:

•? Should we consider any F or D grades (based on Data), i.e., does it significantly reduce the final cost?


71
76
81
86
91
96


71
76
81
86
91
96