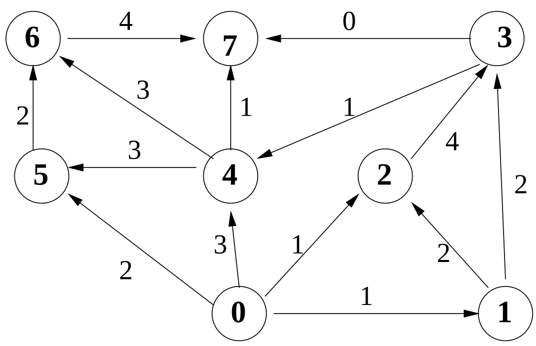
2.6 Shortest paths with nonnegative costs

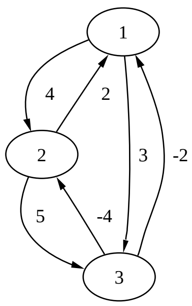
Given the following graph, find a set of shortest paths from node 0 to all the other nodes, using Dijkstra's algorithm.

Can we apply Dynamic Programming to solve the problem. If so, apply it.



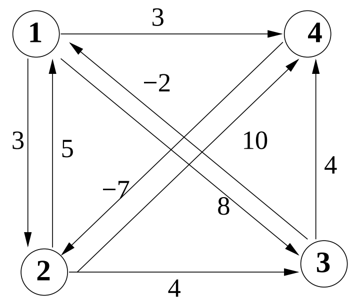
2.7 Shortest paths with negative costs

Given the following graph, find the shortest paths between all pairs of nodes, or show that the problem is ill-posed, by exhibiting a circuit of total negative cost. Apply Floyd-Warshall's algorithm.



2.8 Shortest paths with negative costs and ill-posedness

Given the following graph, find the shortest paths between all pairs of nodes, or show that the problem is ill-posed, by exhibiting a circuit of total negative cost, using the Floyd-Warshall algorithm.



2.9 Shortest paths application

A company buys a machine for 12k Euro. The maintenance costs and the expected revenue for the next 5 years are indicated in the following table.

years	maintenance (kEuro)	recovery (kEuro)
0	2	-
1	4	7
2	5	6
3	9	2
4	12	1

To avoid a high maintenance cost of an older machine, the machine can be replaced at the beginning of the second, third, fourth, and fifth year with a new one.

Show how the problem of determining an optimal maintenance/replacement policy of minimum total cost (price + maintenance - recovery) can be solved via Dynamic Programming.

Hint: Reduce the problem to a minimum cost path problem in an appropriate acyclic graph.

2.10 Project planning

The preparation of the apple pie has long been a tradition for the Smiths family. First the weight of the ingredients has to be determined: flour, sugar, butter, eggs, apples, cream. Then, we must melt down the butter and add it to a mixture of flour, sugar, and eggs. Apples must be added to this new mixture, once they have been peeled and cut into thin slices. The mixture can then be cooked, in the already heatened oven. It is adviced to whip the cream only after the apple slices have been added to the mixture. Once the cake is cooked, the cream is used to garnish it.

The following table reports the time needed for each activity.

Activity		Time (minutes)
A	Weight the ingredients	5
B	Meld the butter down	3
C	Mix flour, eggs, and sugar	5
D	Peel and cut the apples into slices	10
E	Heaten the oven	20
F	Add butter to the mixture	8
G	Add apples to the mixture	4
H	Cook the mixture in the oven	40
I	Whip the cream	10
L	Garnish	5

Draw the graph of the priorities, determine the Tmin and Tmax times by which each activity must begin and end, so as to minimize the time by which the whole project is completed, and identify the critical path. Draw the 'at the earliest' Gantt diagram.

SOLUTION

2.6 Shortest paths with nonnegative costs

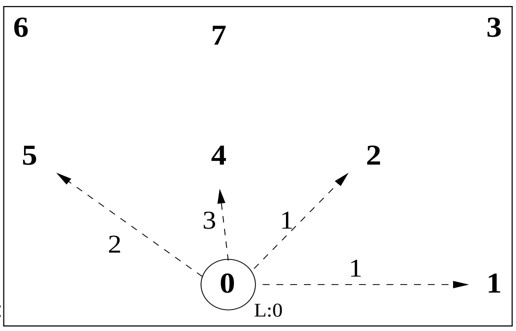
The Dijkstra algorithm determines all the shortest paths from a given node s to all the other nodes in the graph. It can be applied to any (directed) graph, even if containing cycles, with nonnegative edge costs.

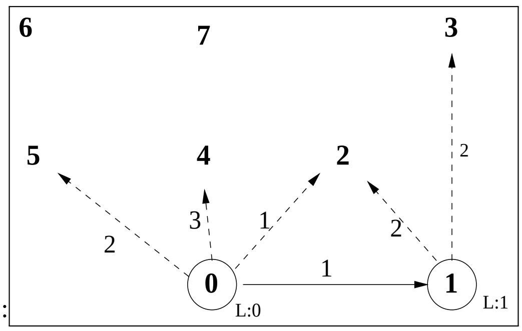
- *Data structures*: S (subset of nodes with fixed labels), initialized to $\{s\}$; $L(i)$ (cost of the shortest path from s to i), initialized to $L(s) = 0$; $p(i)$ (predecessor of node i in the shortest path from s to i), initialized to $p(s) = s$.
- *Description*: at each iteration, from the cut induced by S, we identify an arc (v, h), such that $L(v) + c_{vh}$ is minimum (c_{vh} is the cost of the arc (v, h)). $S \leftarrow S \cup \{h\}$, $L(h) \leftarrow L(v) + c_{vh}$, $p(h) \leftarrow v$. The algorithm halts when $S = V$.

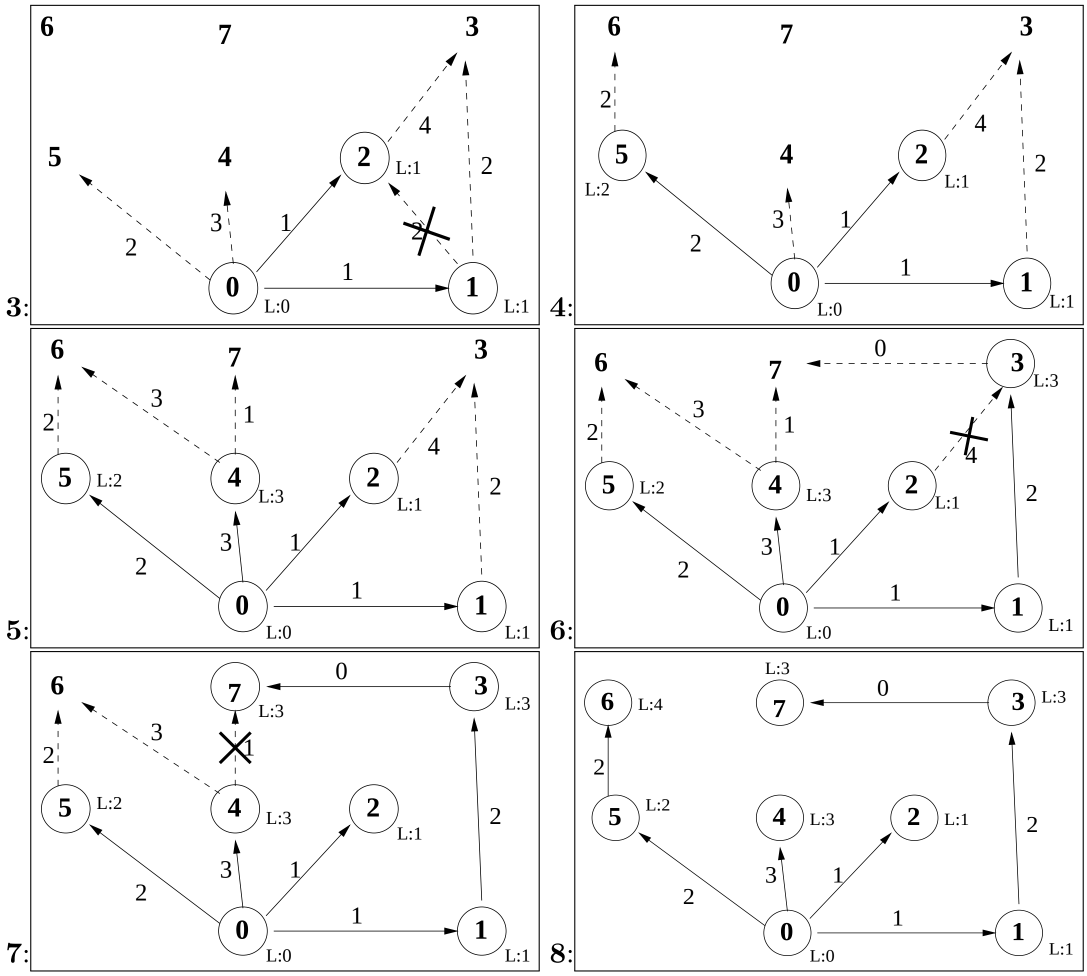
Iterations:

- Initialization: $S = \{0\}, L(0) = 0, p(0) = 0$;
- $(v, h) = (0, 1)$, $L(1) = 1$, $p(1) = 0$, $S = \{0, 1\}$;
- $(v, h) = (0, 2)$, $L(2) = 1$, $p(2) = 0$, $S = \{0, 1, 2\}$;
- $(v, h) = (0, 5)$, $L(5) = 2$, $p(5) = 0$, $S = \{0, 1, 2, 5\}$;
- $(v, h) = (0, 4)$, $L(4) = 3$, $p(4) = 0$, $S = \{0, 1, 2, 4, 5\}$;
- $(v, h) = (1, 3)$, $L(3) = 3$, $p(3) = 1$, $S = \{0, 1, 2, 3, 4, 5\}$;
- $(v, h) = (3, 7)$, $L(7) = 3$, $p(7) = 3$, $S = \{0, 1, 2, 3, 4, 5, 7\}$;
- $(v, h) = (5, 6)$, $L(6) = 4$, $p(6) = 5$, $S = G$: STOP;

We can plot the behaviour of the algorithm as follows. In the pictures, a node is highlighted if it is in S, while an arc is represented with a dashed line if it can be selected in the current iteration, and in a solid line when it is chosen. The labels on the nodes in S indicate the cost of the shortest path from node 0. When a dashed arc is incident to two nodes in S, it is removed. This way, the dashed arcs always belongs to the cut induced by S. The arc (i, j) that is selected is always that with minimum cost $L(i) + c_{ij}$.

1:


2:




A topological order can be introduced as follows. At iteration i, for $i = \{1, \ldots, n\}$, pick a node with no incident arcs, label it as 'node i', remove it from the graph, iterate until all nodes are removed, or there is no node with no incident arcs –in this case the graph is not acyclic.

The graph we are dealing with is acyclic, and its node indices already induce its topological order. The Dynamic Programming technique can therefore be applied. It has complexity $O(m)$, smaller than that of the Dijkstra algorithm of $O(n^2 + m)$.

- $L(0) = 0$, $p(0) = 0$;
- $L(1) = L(0) + c_{01} = 1$, $p(1) = 0$;
- $L(2) = \min\{L(0) + c_{02}, L(1) + c_{12}\} = \min\{0 + 1, 1 + 2\} = 1$, $p(2) = 0$;
- $L(3) = \min\{L(1) + c_{13}, L(2) + c_{23}\} = \min\{1 + 2, 1 + 4\} = 3$, $p(3) = 1$;
- $L(4) = \min\{L(0) + c_{04}, L(3) + c_{34}\} = \min\{0 + 3, 3 + 1\} = 3$, $p(4) = 0$;
- $L(5) = \min\{L(0) + c_{05}, L(4) + c_{45}\} = \min\{0 + 2, 3 + 3\} = 3$, $p(5) = 0$;
- $L(6) = \min\{L(4) + c_{46}, L(5) + c_{56}\} = \min\{3 + 3, 2 + 2\} = 3$, $p(6) = 5$;
- $L(7) = \min\{L(3) + c_{37}, L(4) + c_{47}, L(6) + c_{67}\} = \min\{3 + 0, 3 + 1, 4 + 4\} = 3$, $p(7) = 3$;

2.7 Shortest paths with negative costs

This algorithm finds a shortest path for all pairs of nodes in a (directed) graph with arc costs unrestricted in sign. The graph can contain cycles, but such cycles have to be of nonnegative cost. It has complexity $O(n^3)$, where n is the number of nodes in the graph.

- *Data structures*: let D be a matrix such that element d_{ij} is, at the beginning of the algorithm, the cost of arc (i, j) and, at the end, the cost of a shortest path from i to j. Let P be a matrix of precedecessors ($P = (p_{ij})$, where p_{ij} is the precedecessor of node j in the shortest path from i to j).
- *Description*: at each iteration, we select a node with index $h \leq n$ and compute the length of the paths via h for all pairs of nodes (i, j) different from h[1].

  If the path from i to j, passing through h, is shorter than that the original one from i to j, matrices D and P are updated: d_{ij} is set to the cost of the path through h, and p_{ij} is updated to p_{hj}. The algorithm halts when one of the following conditions holds

  (1) all nodes have been selected for the triangulations

  (2) there is a circuit of negative cost: the problem of finding a shortest path (not necessarily *simple*!) it not well defined. Indeed, it is possible to go through the negative cost circuit an arbitrary number of times, reducing the cost of the path containing such circuit at each iteration. The problem is, evidently, unbounded.

(a) Initial configuration

D	1	2	3
1	0	4	3
2	2	0	5
3	-2	-4	0

P	1	2	3
1	1	1	1
2	2	2	2
3	3	3	3

(b) Iteration $h = 1$ (triangulation on node 1)

$$d_{21} + d_{12} = 2 + 4 = 6 > d_{22} = 0$$
$$d_{21} + d_{13} = 2 + 3 = 5 = d_{23} = 5$$
$$d_{31} + d_{13} = -2 + 3 = 1 > d_{33} = 0$$
$$d_{31} + d_{12} = -2 + 4 = 2 > d_{32} = -4$$

D	1	2	3
1	0	4	3
2	2	0	5
3	-2	-4	0

P	1	2	3
1	1	1	1
2	2	2	2
3	3	3	3

(c) Iteration $h = 2$ (triangulation on node 2)

$$d_{12} + d_{21} = 4 + 2 = 6 > d_{11} = 0$$
$$d_{12} + d_{23} = 4 + 5 = 9 > d_{13} = 3$$
$$d_{32} + d_{23} = -4 + 5 = 1 > d_{33} = 0$$
$$d_{32} + d_{21} = -4 + 2 = -2 = d_{31} = -2$$

D	1	2	3
1	0	4	3
2	2	0	5
3	-2	-4	0

P	1	2	3
1	1	1	1
2	2	2	2
3	3	3	3

(d) Iteration $h = 3$ (triangulation on node 3)

[1] In considering all pairs of nodes i, j, including those where i or j equals h, some useless iterations are performed. Indeed, they amount to check whether a path from i to h is shorter than that from i to h plus that from h to itself

$$d_{13} + d_{31} = 3 - 2 = 1 > d_{11} = 0$$
$$d_{13} + d_{32} = 3 - 4 = -1 < d_{12} = 4$$
$$\Rightarrow \textbf{update } d_{12}, p_{12}$$
$$d_{23} + d_{32} = 5 - 4 = 1 > d_{22} = 0$$
$$d_{23} + d_{31} = 5 - 2 = 3 > d_{21} = 2$$

D	1	2	3
1	0	**-1**	3
2	2	0	5
3	-2	-4	0

P	1	2	3
1	1	**3**	1
2	2	2	2
3	3	3	3

2.8 **Shortest paths with negative costs and ill-posedness**.

(a) Initial configuration

D	1	2	3	4
1	0	3	8	3
2	5	0	4	10
3	-2	∞	0	4
4	∞	-7	∞	0

P	1	2	3	4
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4

(b) Iteration $h = 1$ (triangulation on node 1)

$$
\begin{aligned}
d_{21} + d_{12} = 8 &\quad > \quad d_{22} = 0 \\
d_{21} + d_{13} = 13 &\quad > \quad d_{23} = 4 \\
d_{21} + d_{14} = 8 &\quad < \quad d_{24} = 10 \\
&\quad \Rightarrow \quad \textbf{update } d_{24}, p_{24} \\
d_{31} + d_{12} = 1 &\quad < \quad d_{32} = \infty \\
&\quad \Rightarrow \quad \textbf{update } d_{32}, p_{32} \\
d_{31} + d_{13} = 6 &\quad > \quad d_{33} = 0 \\
d_{31} + d_{14} = 1 &\quad < \quad d_{34} = 4 \\
&\quad \Rightarrow \quad \textbf{update } d_{34}, p_{34} \\
d_{41} + d_{ij} = \infty (\forall i, j)
\end{aligned}
$$

D	1	2	3	4
1	0	3	8	3
2	5	0	4	**8**
3	-2	**1**	0	**1**
4	∞	-7	∞	0

P	1	2	3	4
1	1	1	1	1
2	2	2	2	**1**
3	3	**1**	3	**1**
4	4	4	4	4

(c) Iteration $h = 2$ (triangulation on node 2)

$$
\begin{aligned}
d_{12} + d_{21} = 8 &> d_{11} = 0 \\
d_{12} + d_{23} = 7 &< d_{13} = 8 \\
&\Rightarrow \textbf{update } d_{13}, p_{13} \\
d_{12} + d_{24} = 11 &> d_{24} = 3 \\
d_{32} + d_{21} = 6 &> d_{31} = -2 \\
d_{32} + d_{23} = 5 &> d_{33} = 0 \\
d_{32} + d_{24} = 9 &> d_{34} = 1 \\
d_{42} + d_{21} = -2 &< d_{41} = \infty \\
&\Rightarrow \textbf{update } d_{41}, p_{41} \\
d_{42} + d_{23} = -3 &< d_{43} = \infty \\
&\Rightarrow \textbf{update } d_{43}, p_{43} \\
d_{42} + d_{24} = 1 &> d_{44} = 0
\end{aligned}
$$

D	1	2	3	4
1	0	3	**7**	3
2	5	0	4	8
3	-2	1	0	1
4	**-2**	-7	**-3**	0

P	1	2	3	4
1	1	1	**2**	1
2	2	2	2	1
3	3	1	3	1
4	**2**	4	**2**	4

(d) Iteration $h = 3$ (triangulation on node 3)

$$
\begin{aligned}
d_{13} + d_{31} &= 5 > d_{11} = 0\\
d_{13} + d_{32} &= 8 > d_{12} = 3\\
d_{13} + d_{34} &= 8 > d_{14} = 3\\
d_{23} + d_{31} &= 2 < d_{21} = 5\\
&\Rightarrow \textbf{update } d_{21}, p_{21}\\
d_{23} + d_{32} &= 5 > d_{22} = 0\\
d_{23} + d_{34} &= 5 < d_{24} = 8\\
&\Rightarrow \textbf{update } d_{24}, p_{24}\\
d_{43} + d_{31} &= -5 < d_{41} = -2\\
&\Rightarrow \textbf{update } d_{41}, p_{41}\\
d_{43} + d_{32} &= -2 > d_{42} = -7\\
d_{43} + d_{34} &= -2 < d_{44} = 0\\
&\Rightarrow \textbf{update } d_{44}, p_{44}
\end{aligned}
$$

D	1	2	3	4
1	0	3	7	3
2	**2**	0	4	**5**
3	-2	1	0	1
4	**-5**	-7	-3	**-2**

P	1	2	3	4
1	1	1	2	1
2	**3**	2	2	**3**
3	3	1	3	1
4	**3**	4	2	**1**

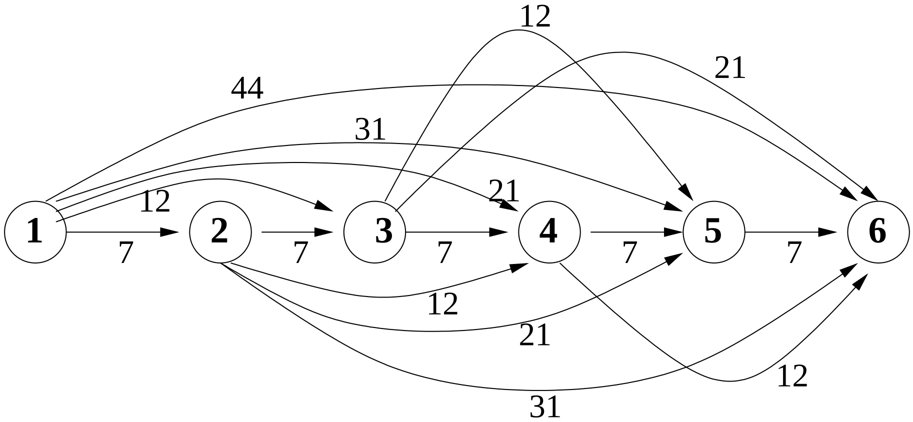
We obtain $d_{44} = -2 < 0$ and the algorithm halts: we found a circuit of total negative cost of -1 ($4 \to 2 \to 3 \to 1 \to 4$).

2.9 **Shortest paths application**

Consider a directed graph with 6 nodes: nodes from 1 to 5 are associated to the beginning of each year, while node 6 indicates the end of the period of interest. For each pair i, j, $i, j = 1, \ldots, 5$, $i < j$, arc (i, j) represents the choice of reselling, at the beginning of year j, a machine that was bought at the beginning of year i. The cost c_{ij} is defined as

$$c_{ij} = a + \sum_{k=i}^{j-i-1} m_k - r_{j-i},$$

where a is the buying cost of 12000 Euro, m_k is the maintenance cost for year k, and r_j is the price at which the machine is sold. We obtain the following directed acyclic graph



Any path from 1 to 6 corresponds to a plan, with a cost equivalent to that of the path.

Therefore, we need to look for a shortest path from 1 to 6. We apply the Dynamic Programming algorithm, obtaining

(a) $L(1) = 0$;
(b) $L(2) = 7$, $p(2) = 1$;
(c) $L(3) = 12$, $p(3) = 1$;
(d) $L(4) = 19$, $p(4) = 3$;
(e) $L(5) = 24$, $p(5) = 3$;
(f) $L(6) = 31$, $p(6) = 5$.

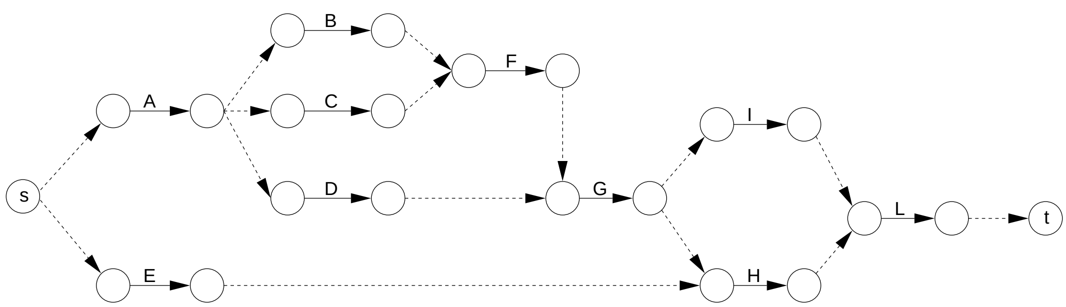
A shortest path (of cost 31) is $1 \to 3 \to 5 \to 6$. It amounts to buying a new machine every 2 years. Note that there are other optimal solution, such as $1 \to 3 \to 4 \to 6$.

2.10 **Project planning**

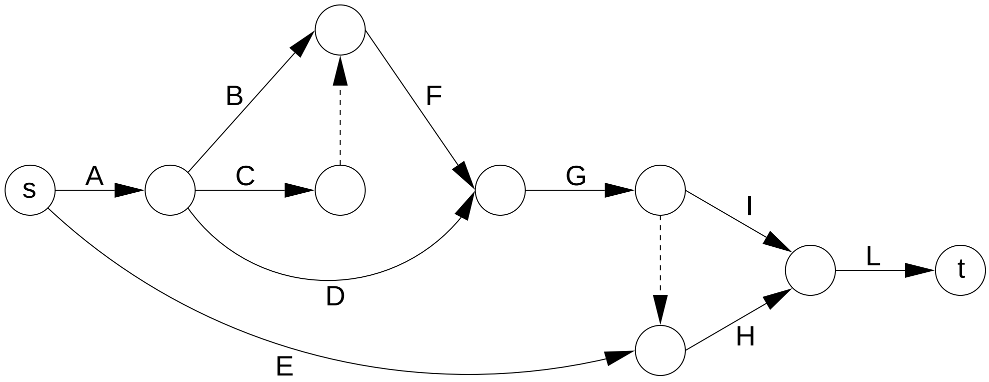
The table indicates lengths and prerequisites for each activity

Activities		Length	Predecessors
A	Weight the ingredients	5	-
B	Meld the butter down	3	A
C	Mix flour, eggs, and sugar	5	A
D	Peel and cut the apples into slices	10	A
E	Heaten the oven	20	-
F	Add butter to the mixture	8	B,C
G	Add apples to the mixture	4	D,F
H	Cook the mixture in the oven	40	E,G
I	Whip the cream	10	G
L	Garnish	5	H,I

We derive the prerequisites graph. For each activity, we introduce two nodes, representing the beginning and the end of the activity, and an arc of cost equivalent to the length of the activity. For each prerequisite relationship $A_i < A_j$, and arc (i, j) is introduced (dashed lines) between the ending node of A_i and the beginning node of j. We add a node s, connected to the beginning node of each activity, with cost 0, for all activities with no prerequisites. Similarly, we add a node t, and connect to it all the ending nodes for all the activities that are not a prerequisite for any other activities, with cost 0.



The graph can be reduced to a smaller one as shown in the lectures. The following graph is obtained.



We use the Critical Path Method (CPM) to determine T_{min} and T_{max} for each node of the graph.

The algorithm is composed of two phases, were the Dynamic Programing technique is applied to find longest and shortest paths in the graph. The algorithm is outlined as follows.

(a) $T_{min}(1) = 0$;

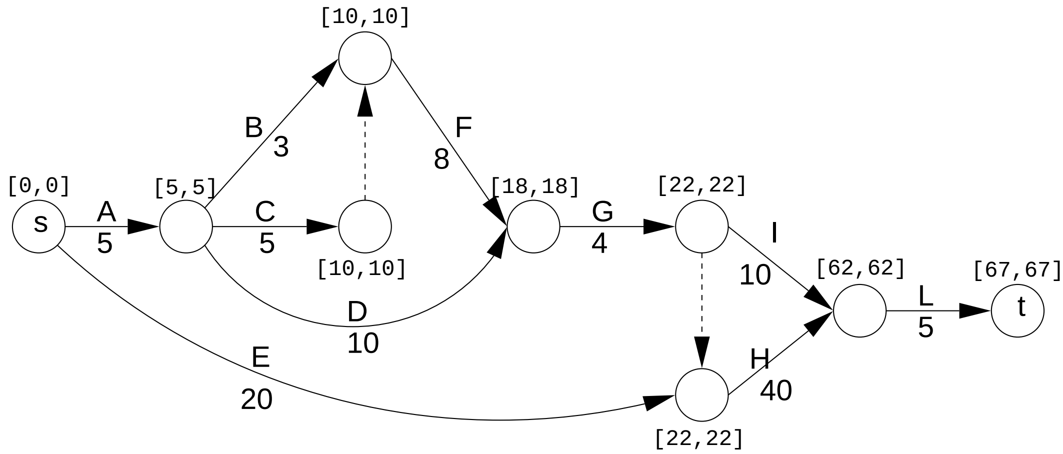
(b) sort the nodes in topological order;

(c) for each $h = 2, \dots, n$:
let $T_{min}(h) = \max\{T_{min}(i) + d_{ih} \mid (i,h) \in \delta^-(h)\}$;

(d) $T_{max}(n) = T_{min}(n)$;

(e) for each $h = n-1, \dots, 1$:
let $T_{max}(h) = \min\{T_{max}(i) - d_{ih} \mid (i,h) \in \delta^+(h)\}$.

We obtain the following solution



The slacks for the activities are

(a) $\sigma(A) = T_{max}(2) - T_{min}(1) - d_{12} = 5 - 0 - 5 = 0$

(b) $\sigma(B) = T_{max}(4) - T_{min}(2) - d_{24} = 10 - 5 - 3 = 2$

(c) $\sigma(C) = T_{max}(3) - T_{min}(2) - d_{23} = 10 - 5 - 5 = 0$

(d) $\sigma(D) = T_{max}(5) - T_{min}(2) - d_{25} = 18 - 5 - 10 = 3$

(e) $\sigma(E) = T_{max}(6) - T_{min}(1) - d_{16} = 22 - 0 - 20 = 2$

(f) $\sigma(F) = T_{max}(5) - T_{min}(4) - d_{45} = 18 - 10 - 8 = 0$

(g) $\sigma(G) = T_{max}(6) - T_{min}(5) - d_{56} = 22 - 18 - 4 = 0$

(h) $\sigma(H) = T_{max}(8) - T_{min}(7) - d_{78} = 62 - 22 - 40 = 0$

(i) $\sigma(I) = T_{max}(8) - T_{min}(6) - d_{68} = 62 - 22 - 10 = 30$

(j) $\sigma(I) = T_{max}(9) - T_{min}(8) - d_{89} = 67 - 62 - 7 = 0$

The critical activities are A, C, F, G, H, L.

The 'at the earliest' Gantt diagram is