10\1\12

Critical Path Analysis

Using a network model to control progress of a project

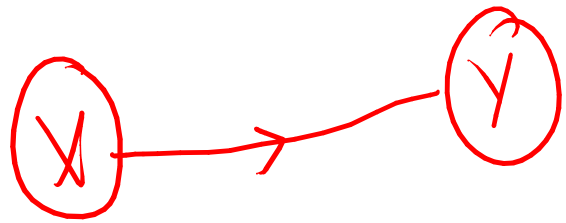
Project = { activities }

vertices

Relationships between activies:

Must complete X before starting Y

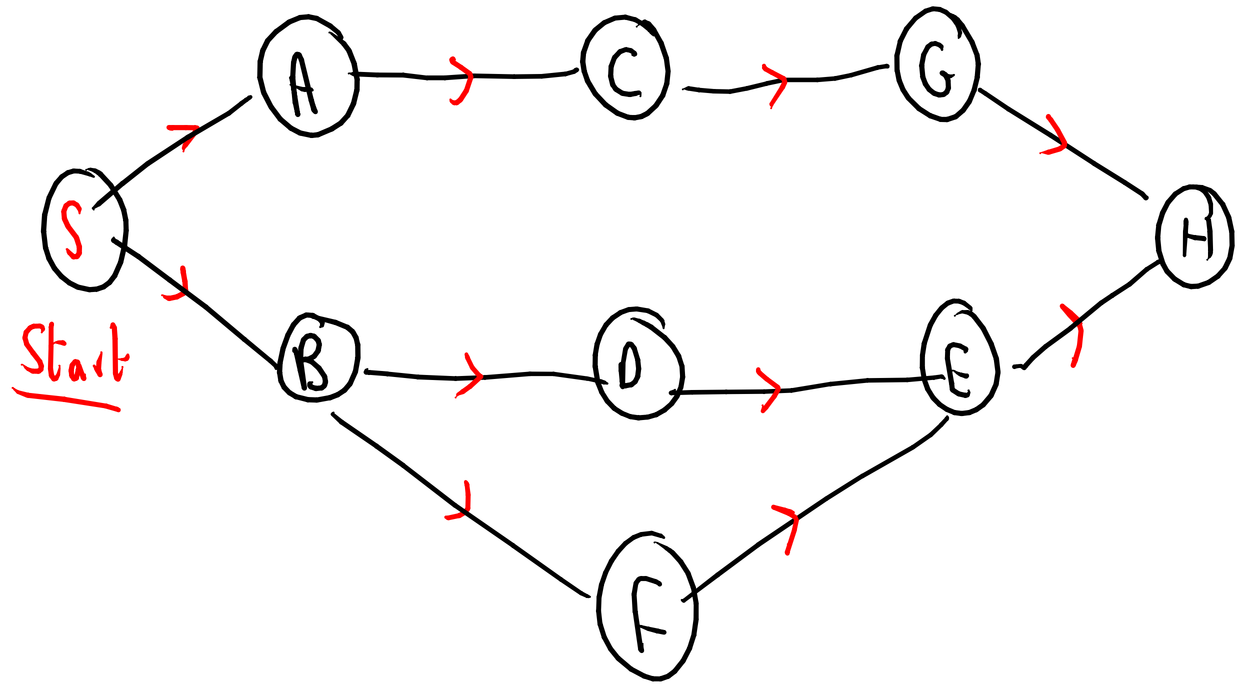
Digraph:



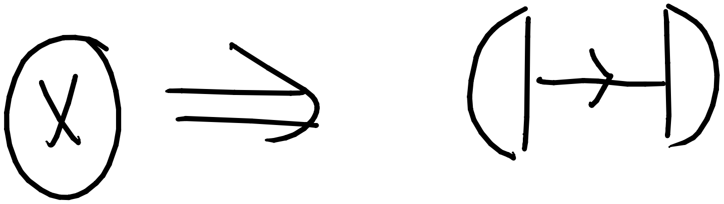
Each activity has a duration.

Question: what is minimum time needed to complete project

Example 8 activities

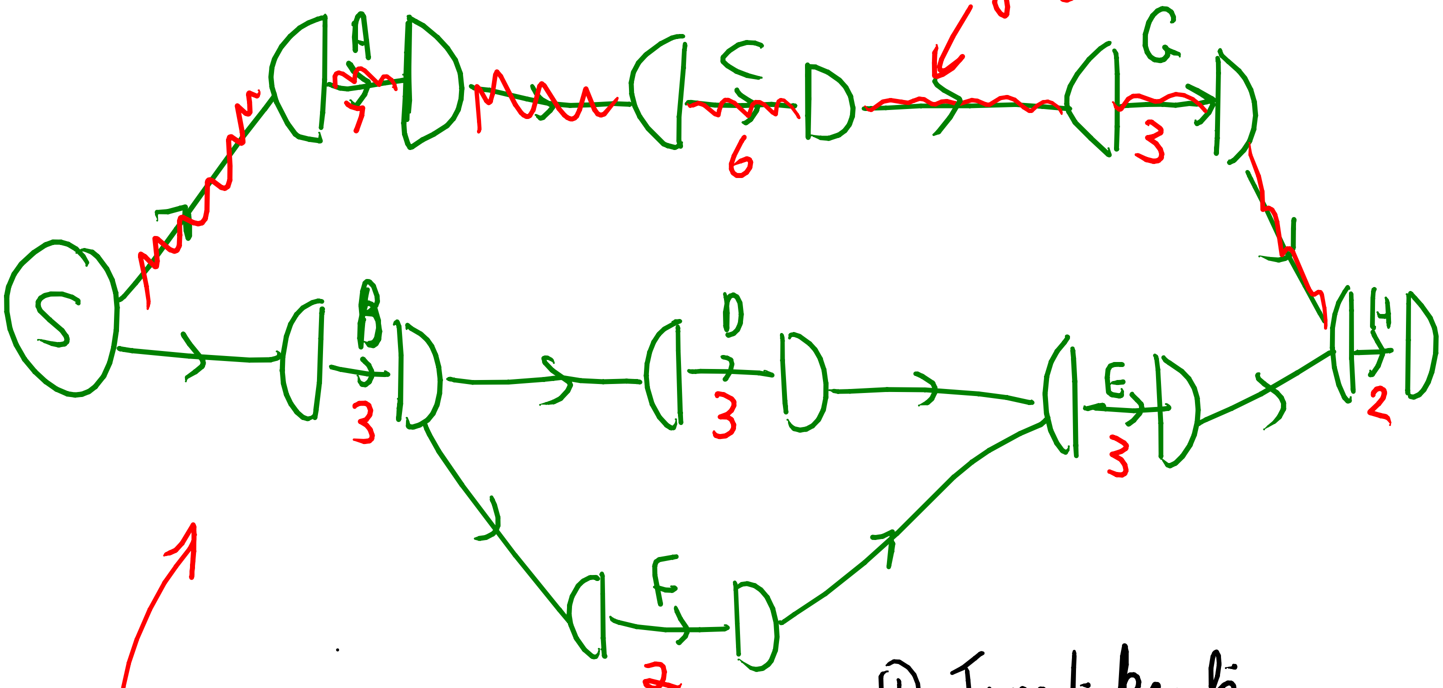


Activity	A	B	C	D	E	F	G	H
Duration	7	3	6	3	3	2	3	2

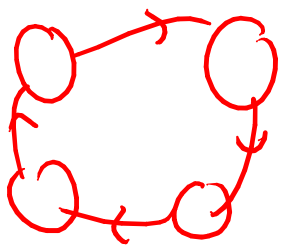


Longest path ≡ critical path

length 0



DAC



① Time taken to complete project = maximum length

② a path from S .... D
H

Float of an activity = Latest start time − Earliest start time

(assuming we finish a.s.a.p.)

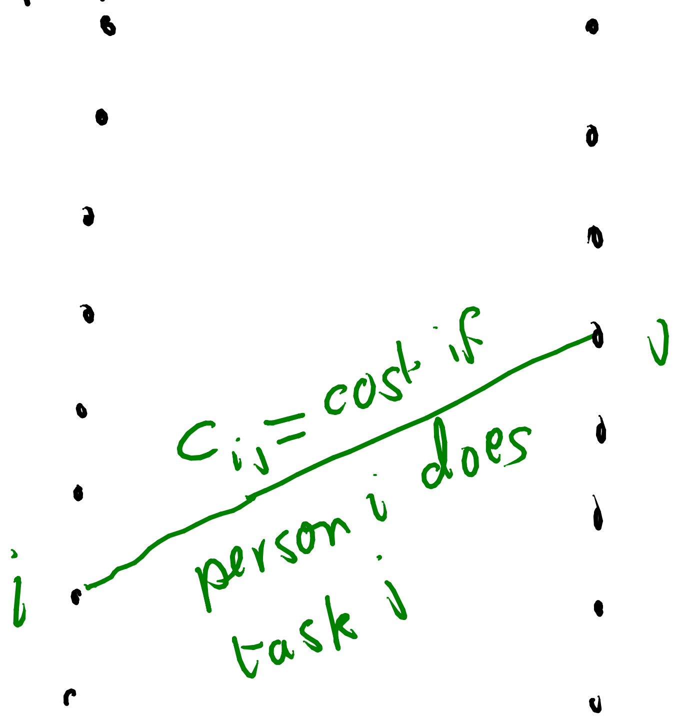
= 0 for critical activities

Suppose T = finish time.

= T − (max. distance to Finish from ( )) − max. distance from start to ( )

Assignment Problem

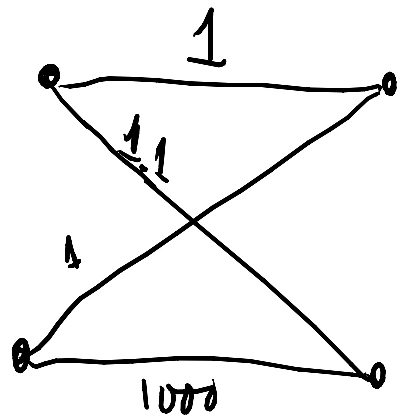
m people $\qquad$ m tasks



Each person is assigned a different task

solutions = $m!$

Problem: assign tasks to minimise total cost



Being "myopic" is not always a good idea.

Idea

For $k = 1$ to m do

$$\left\{ \begin{array}{l} \text{given min. cost assignment of} \\ [k] \longrightarrow [k], \\ \\ \text{find min. cost assignment of} \\ [k+1] \longrightarrow [k+1] \end{array} \right.$$

SHORTEST PATH PROBLEM

10\3\12

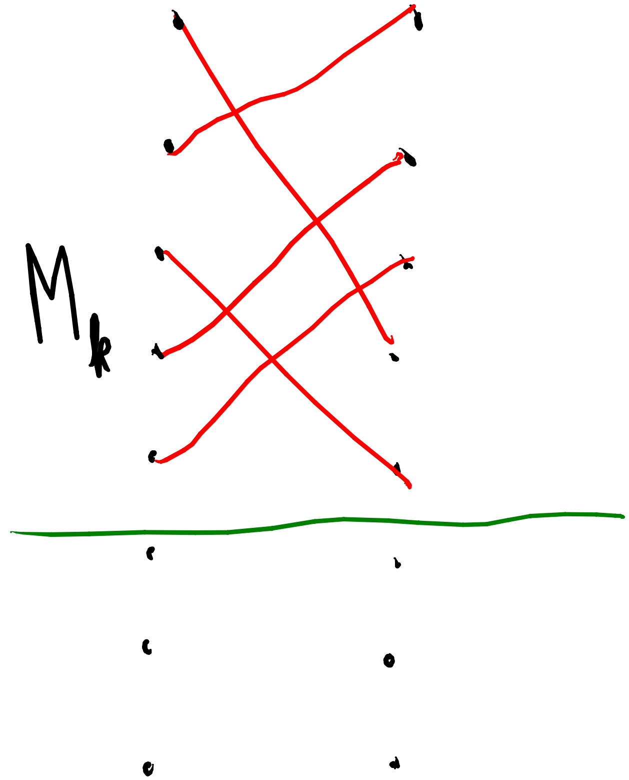
Suppose we have min. cost assignment

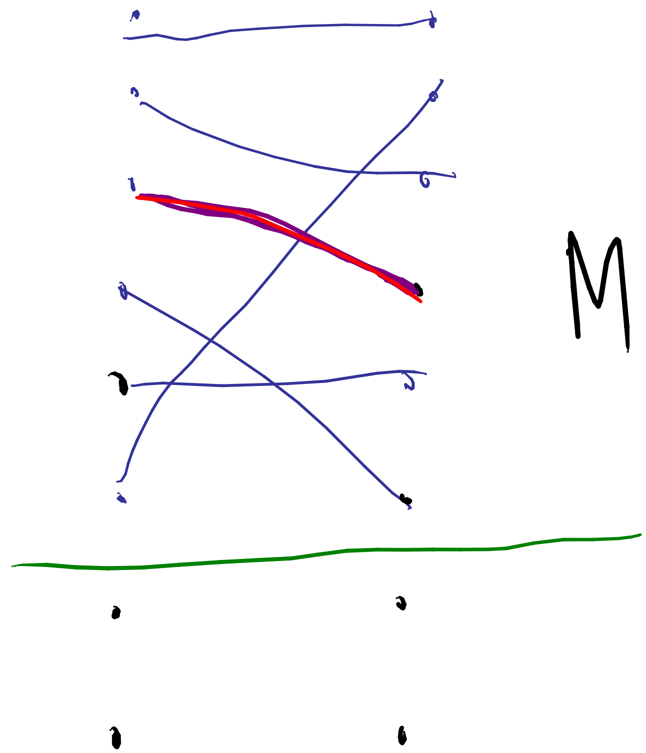
$g: [k] \to [k]$.

How do we get min. cost assignment

of $[k+1] \to [k+1]$?

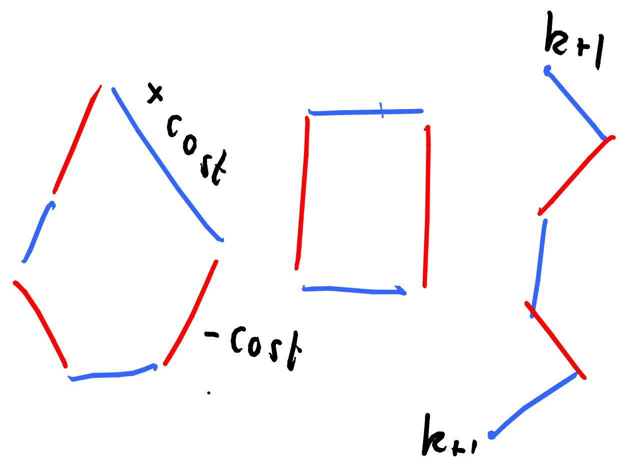
min. $[n] \to [k]$

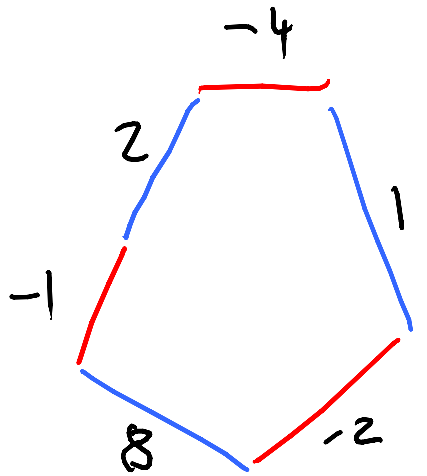


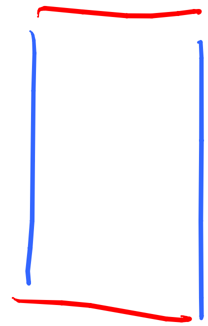


$M \oplus M_k$

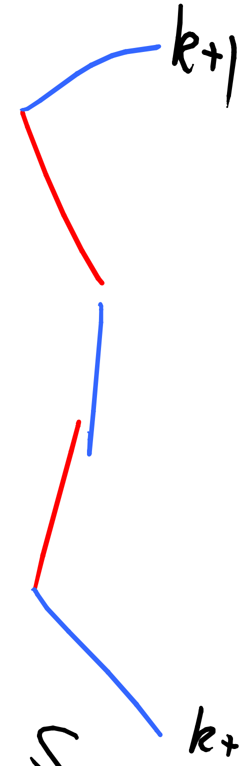
$(M \setminus M_k) \cup (M_k \setminus M)$







Same here



Same here

$8+1+2-(2+4+1)$

= change in cost

if I replace

by

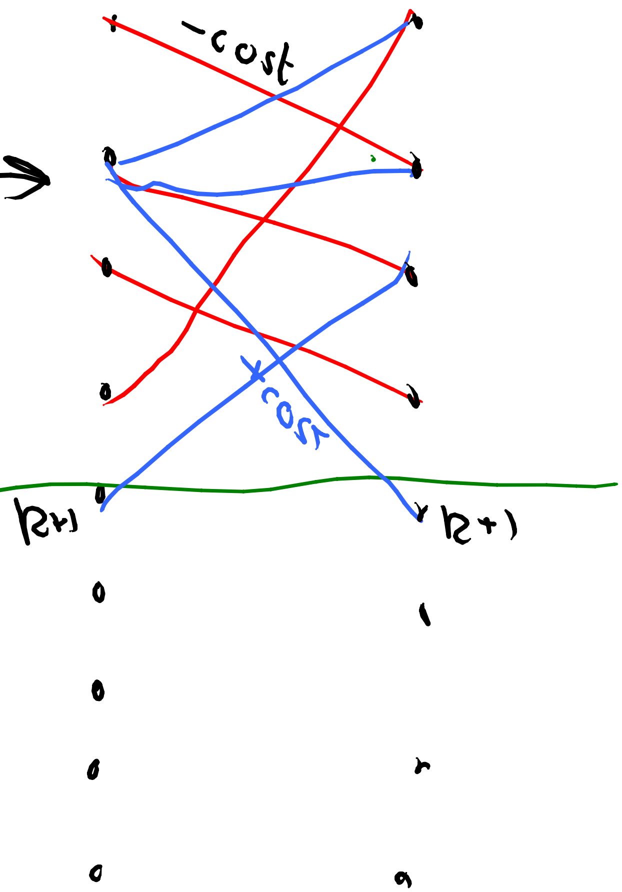
For a cycle sum of red costs $\geq$ sum of blue costs

else M_k is not minimal

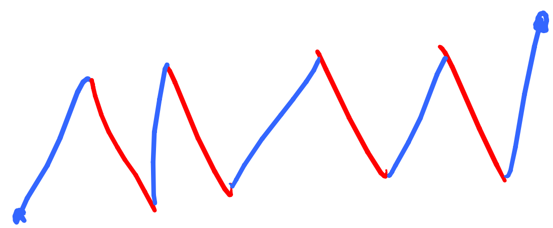
Therefore choose M with no cycles & shortest length

Algorithm

only some edges, to avoid mess



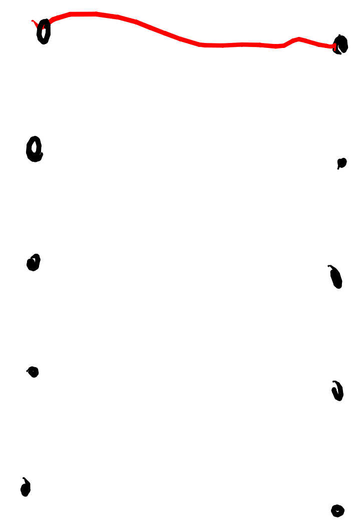
Find shortest path from (k+1) to (k+1)



Add and

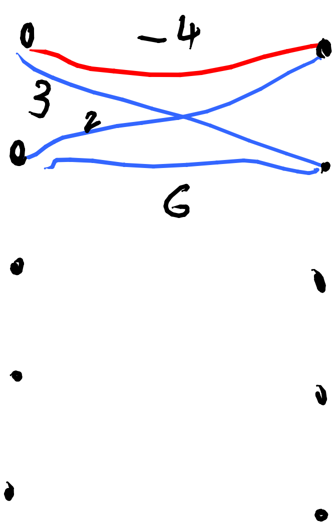
delete

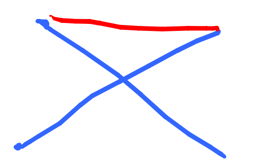
$$\begin{bmatrix} 4 & 3 & 1 & 4 & 5 \\ 2 & 6 & 4 & 3 & 1 \\ 3 & 2 & 1 & 5 & 6 \\ 1 & 4 & 3 & 2 & 5 \\ 2 & 3 & 1 & 4 & 2 \end{bmatrix}$$



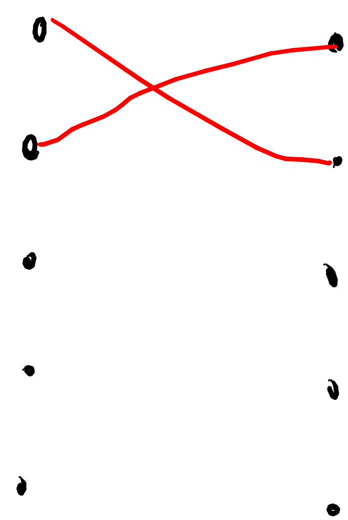
$k=1$

$k=2$



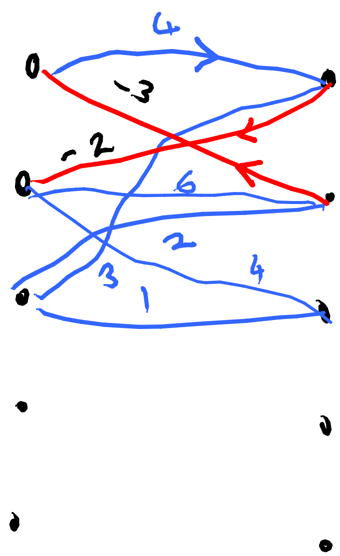


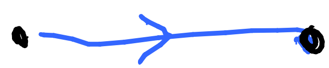
Shortest



$$\begin{bmatrix} 4 & 3 & 1 & 4 & 5 \\ 2 & 6 & 4 & 3 & 1 \\ 3 & 2 & 1 & 5 & 6 \\ 1 & 4 & 3 & 2 & 5 \\ 2 & 3 & 1 & 4 & 2 \end{bmatrix}$$

$k = 2$





Shortest

Total running time

$$= O(n \times n^3) = O(n^4)$$

Linear Programming

Minimise $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

s.t.

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i$$

Every basic feasible solution to LP with $0 \leq x_{ij} \leq 1$ satisfies $\boxed{x_{ij} = 0 \text{ or } 1}$

replace by

$x_{ij} = 1$