Project planning

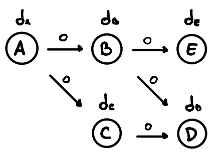
Project : set of m activities $A_1, \dots, A_m$ each with a known duration d_j
with a set of predences $A_i < A_j$

Obj : find the min makespan → termination of the last activity

Possible representations :

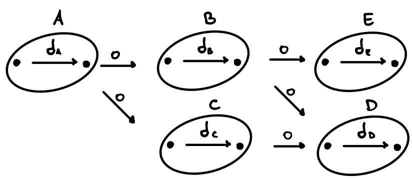
Activities : A, B, C, D, E

Precedences : $A<B$, $A<C$, $B<D$, $B<E$, $C<D$

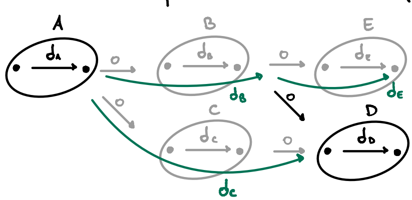


→ Node = activity → The cost is on the node
Arc = precedence and not in the arc

We could use another representation to have a more standard graph → Arc = Activity



Paths that have no ramifications could be simplified:



Critical path metod

For this method we will consider only graphs that are:

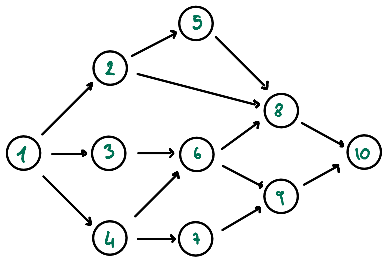
- acyclic
- with only one start node → ceate a dummy node and make the new common start
- in topological order → If the graph is acyclic, there is a topological order

Finding the topological order

Idea:

- find the start node and add it to the queue
- remove that node
- repeat

$\Big\}\ O(m)$



After finding the longest path, we can set it as lower-bound: longest path ≤ min makespan

CPM (Critical path method)

Compute for each node $T_{min\,h}$ and $T_{max\,h}$

$T_{min,1} \leftarrow 0$

for $h \leftarrow 2$ to n do:

$T_{min\,h} \leftarrow \max \{ T_{min\,i} + d_{i,h} \mid (i,h) \in \delta^-(h) \}$

(Similarly for T_{max})

$T_{min\,h}$ = The earliest I could ever start activity h without breaking any preferences

$T_{max\,h}$ = The latest I could start an activity h without causing a delay to the whole project

Nodes where $T_{min,h} = T_{max,h}$ are called critical nodes

↓

the path through the critical nodes is the critical path