Suppose $D$ in a OAS.

Observation: $D$ has at least one sink (a vertex with no outgoing edge).

Let $P$ be a longest path in $D$. If $P$ would not be a cycle,
then $P$ would not be longest.

Algorithm:
1. Find a sink $v_a$.
2. Topologically order $D - v_a$.

$D$ would not be a cycle if $v_a$ is a sink.

1 by induction.
Acyclic Digraphs (DAG)

A DAG is a digraph with no directed circuits.

Topological Ordering

$D=(V,E)$ is a digraph

$V=\{v_1, v_2, \ldots, v_n\}$

topological ordering $\pi (v_i, v_j) \in E \implies i < j$

Theorem

A digraph $D$ has a topological ordering iff it is a DAG.

Proof

Suppose $v_1, v_2, \ldots, v_n$ is a top. ord.

Suppose $C=(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$ is a directed cycle.

$\implies i_1 < i_2 < \ldots < i_k < i_1$, Contradiction
Acyclic Digraphs (DAG)

A DAG is a digraph with no directed circuits.

Topological Ordering

D=(V,E) is a digraph
V={v1,v2,..,v3}

If (v_i,v_j) E \implies i < j

To find a longest path from v_i to every other vertex:

\[ d(v_i) = \max_{j} d(v_j) + d(v_{i,j}) \]

\( d(v_j) \) is correct. Induction on \( j \).

\( j = 1 \) is trivial.

longest path

...
Critical Path Analysis

Large project is broken into activities

Each activity has a duration

We have a digraph $D = (\{\text{activities}\}, E)$ such that $\text{length}(e) = \text{duration}(e)$.

Also if $\exists \delta_0 \rightarrow \delta_1 \rightarrow \cdots \rightarrow \delta_n$

you can't carry out $\delta_n$ until $\delta_0$ is complete.

Also if $\exists \delta_0 \rightarrow \delta_1 \rightarrow \cdots \rightarrow \delta_n$

Length of longest path from Start to Finish = minimum time needed for project.
Assignment Problem

\[ \pi(i) : \pi(i) \text{ is a permutation.} \]

\[ \text{Minimise } C(\pi) = \sum_{i=1}^{n} c(i, \pi(i)) \]

\( C(i) = \text{cost of completing task } i \)

If it is done by worker \( i \)

Problem: Assign tasks to people to minimise total cost:

Graph Theory Outline:
- We have a complete bipartite graph \( G \)
- A matching is a set of disjoint edges
- Each edge has a cost and cost of matching \( M = \sum_{e \in \pi} c(e) \)
- A matching is perfect if every vertex of \( G \) lies on some edge.
- Find minimum cost perfect matching
A path $P = (e_1, e_2, \ldots, e_{2k})$ is alternating with respect to matching $M$ if

$$e_1, e_2, \ldots, e_{2k} \in M \ \text{or} \ \overline{e_1}, \overline{e_2}, \ldots, \overline{e_{2k}} \in M$$

Augmenting if it begins and ends at uncovered vertices.