Directed $G$.

$OPT(i) =$ length of longest path ending at $i$.

$$OPT(i) = \max_{j: j \rightarrow i} OPT(j) + 1.$$ 

$OPT(i, l) =$ says it thru a path of length $l$ ending at $i$.

$$OPT(i, l) = \begin{cases} 
\text{true} & \text{for all } j: j \rightarrow i \\
& \text{if } OPT(j, l-1) = \text{true} \\
& \text{then } OPT(i, l) = \text{true} \\
& \text{and else } \text{false} 
\end{cases}$$

Wrong BC I could have visited $i$ to get to $j$.

$OPT(i, S) =$ longest path ending at $i$ using vertices in a set $S$ of vertices.

Exponentially many states.

Now assume DAG.

OBS: if $j \rightarrow i$, $i$ cannot be in the longest path ending at $j$. O/w we get a cycle.
\[\text{OPT}(i) = \max_{j \text{ s.t. } j \rightarrow i} \text{ path ending at } i.\]
\[\text{OPT}(i) = \max_{j \rightarrow i} \text{ OPT}(j) + 1.\]

**What order?** The topological order.

To make sure we compute \(\text{OPT}(j)\) before \(\text{OPT}(i)\).

Alternatively, \(\text{OPT}(i)\):
- \(\text{OPT}(i) = 0.\)
- For \(j = 1 \ldots n\)
  - If \(j \rightarrow i\)
    - Then \(\text{OPT}(i) = \max(\text{OPT}(i), \text{OPT}(j) + 1)\)

\[a_1, a_2, \ldots, a_n\]

\(\text{if } a_i < a_j\)

**Note:** This is a DAG

(i) Every path corresponds to an increasing subseq

So longest path = longest increasing subseq.

**Shortcut path with my edge weights**

\[\text{OPT}(i) = \text{ shortest path ending at } i.\]
\[\text{OPT}(i) = \min_{j \rightarrow i} \text{ OPT}(j) + c_{ji}.\]
Problem: What is the right order? Is \( j \) closer to \( s \) or \( i \) is closer?

\[
\text{OPT}(i, l) = \text{shortest path ends at } i \text{ which has } \leq l \text{ edges.}
\]

\[
\text{OPT}(i, l) = \min_{j : j \to i} \text{OPT}(j, l-1) + c_{j,i}.
\]