Dynamic Programming

EDIT Distance
- Optimal triangulation
- Independent set tree
- Longest increasing sequence
- Connection to DAGs.
**DAG**

Topological sorting

Sink

---

**Longest Increasing Sequence**

\[ x_1, x_2, \ldots, x_n \]

\[ \begin{array}{c}
7, 1, 3, 5, 3, 1, 6, 4, 7, 3, 3, 7, 4 \\
\end{array} \]

---

**Dilworth Theorem**

Given a sequence of n numbers, there is always a subseq at length \( m \) that is either increasing or decreasing.

---

\[ f(i) = \text{length of longest increasing sequence that has } x_i \text{ as last element} \]

\[ \max_{i \leq n} f(i) = O(n) \]
Data structure magic

Binary search trees

\[ \text{STREPS} \]

\[ O(\log n) \leq \]

\[ \text{Entry: } (\text{key, value}) \]

sorted by the keys

max value in the subtree

\[ f(i) \text{ - length of longest increasing sequence ending with } x_i \]

\[ f(n) = \max_{x_j < x_n} (f(j) + 1) \]
\[ O(\log n) \leq O(n^2) \]

- Sep 6-3:10 PM

- Sep 6-3:11 PM

- Sep 6-3:14 PM

- Sep 6-3:15 PM