Algorithm Analysis and Design (CS1.301)

Monsoon 2021, IIIT Hyderabad
18 September, Saturday (Lecture 9)

Dynamic Programming

Dynamic programming enables us to store the results of intermediate computations in case some sub-problems partially overlap and need the same results. Thus, we do not need to make the same calls repeatedly.

Shortest Path in a DAG

When we need to find the distance from a node s to all the other nodes, we use the following algorithm:

```
dist(s) = 0
for each v \in V - {s} in linearised order:
    dist(v) = min (dist(u) + l(u,v))
                over all (u,v) \in E
```

Here, we can store the values of `dist` for each vertex as they are computed, and thereby avoid making the calls repeatedly.

Longest Increasing Subsequence (LIS)

Given a sequence of numbers $a_1, \ldots, a_n$, we need to find the values of $i_1, \ldots, i_k$ such that $1 \leq i_1 < \cdots < i_k \leq n$ and $a_{i_1} < \cdots < a_{i_k}$ for the largest value of k.

Given an instance of the problem, we will construct a DAG by establishing a node i for every element a_i, and adding directed edges (i, j) whenever $i < j$ and $a_i < a_j$.

Now we can find the LIS by toposorting the DAG and finding the longest path in it. The algorithm for this is exactly similar to that for the shortest path.