Lecture 9

Gidon Rosalki

2024-11-27

1 Dynamic algorithms

Example 1 (Task assignment)**.** *Something is being produced in a factory. There are 2 equivalent production lines, where for both of them there are n work stations. The item can be on one of the lines at once. For every movement between work stations, there is a price. We are looking for the production process of minimal price. (So the input is a directed, weighted graph, with two main lines from a start node S, to an end node F, with lines between the nodes between the lines as well)*

Input: A directed weighted graph with $2n+2$ nodes, and $4n$ edges. (where there are 2 main paths u, and d from S to F, made of nodes $u_1, \ldots, u_n$ and $d_1, \ldots, d_n$ respectively, and for every $i = j - 1$ there is also a link between u_i to d_j, and d_i to u_j). Also a collection of weights of the edges, where $x_1, \ldots, x_{n+1}$ which is the upper line, $y_1, \ldots, y_n$ the lower line, $a_2, \ldots, a_n$ from the upper line to the lower line, and $b_2, \ldots, b_n$ from the lower line to the upper.
Output: The path with the lowest price in the graph from S to F, where the price of the path is the sum of the prices of the edges.

Solution. We may use Dijkstra in $O(n \log(n))$, but instead we shall use a dynamic algorithm to solve this.

The collection of correct solutions $S = \{\text{All the paths between } S \text{ and } F \text{ in the graph}\}$, there are 2^n such paths, and there is a price function on the space, defined by the price of the path. We are looking for the path with the minimal price.

We may use Bellman's principle, where every sub solution of an optimal solution is also optimal.
Our first division will be according to the first decision, to travel from S to u_1, or from S to d_1. So we must first find the optimal path from, u_1 to F, and from d_1 to F in order to know which choice to make. Note that from u_1 and d_1, the sub problems in each case are from u_2 and d_2, so these each only need calculating once. Every sub problem that we will see are of the following design: To travel at a minimal price from some node to F
We will write p^* to be the minimal price of the path from S to F. $p_u(k)$ to be the minimal price from u_k to F, and $p_d(k)$ to be the minimal price from d_k to F. Note that $1 \leq k \leq n$. Overall we have $2n+1$ sub problems. (x_1 is the price from S to u_1, and y_1 to d_1) Recursion formula:
The first recursion formula: $p^* = \min p_u(1) + x_1, p_d(1) + y_1$
General recursion formula: $p_u(k) = \begin{cases} k = n, & \text{if } x_{n+1} \\ kMn, & \text{if } \min p_(k+1) + x_{k+1}, p_d(k+1) + a_{k+1} \end{cases}$

□