CentraleSupélec

Algorithmics and complexity

TD 4/7 – Dynamic Programming

The goal of this TD is to model problems and solve them using dynamic programming.

This TD is 3h and contains practice sessions in Python.

Exercise 1 : Trading policy

We have one Euro that we want to invest in an optimal way for 10 periods. We know exactly the gain factor c_i for an investment at the beginning of period i.

Period i	1	2	3	4	5	6	7	8	9	10
c_i	2	4	1	2	6	2	2	4	1	4

At the beginning of each period, we have two choices: we can either invest the totality of the available money, or keep the money and invest nothing. Let's assume that at the beginning of period i we have an amount of x. If we do not invest any money for period i, we will have x available at the beginning of period $i+1$. However, if we invest x at the beginning of period i, we will get $c_i x$, but only at the end of period $i+1$, so the money will not be available during two periods, and it will be possible to invest it again only at the beginning of period $i+2$.

We are looking for an investment policy which gives us the most money at the beginning of period 12 (so there will not be any investment at the beginning of period 11).

Question 1

By first ignoring the gain coefficients, model the problem by an non-weighted graph (the vertices and edges) where each total investment policy will correspond to a path on this graph. What is the nature of this graph? Draw the graph.

Question 2

If we want to get back to the the shortest path problem, what weights should be assigned to the arcs? Complete the graph.

Question 3

What algorithm seen in class can be used to respond to the exercise? Find the solution of the numerical example.

Exercise 2 : Ski Rental

Decision Problem

A ski shop has n pairs of skis. There are m customers who wish to rent skis, with $m < n$. The shop owner wants to optimize the comfort of skiers. He wants to assign skis to skiers so that the sum of the absolute differences of the heights of each skier and his skis is minimized.

Question 1

Formalize this optimization problem.

Question 2

Consider a greedy algorithm which allocates skis using the following method: we seek for the couple (ski, customer) having the smallest absolute difference, we allocate this pair of skis to that customer and we start again with remaining skis and customers.

Write the Python code of this algorithm. To help you, go to the practice page of the TD:

Question 3

What is the complexity of this algorithm?

Question 4

Does the greedy algorithm give the optimal solution?

Dynamic Programming

We assume without loss of generality that the n pairs of skis and the m customers are sorted in ascending order of size. We note $Sol[i, j]$ the cost of the optimal solution (i.e. the sum of size differences) for the i first pairs of skis and the j first customers. We want to apply the dynamic programming approach to find the cost of the optimal solution $Sol[n, m]$.

We observe that for two pairs of skis and 2 customers, it is better to allocate the smallest pair to the shortest customer and the longest pair to the tallest one. So if we generalize this result to i pairs of skis and j customers, when the first $j - 1$ customers are already served, the customer j can choose his pair of skis and we will not find a better solution by swapping his pair with a smaller customer (served before him).

This observation allows us to define a resolution algorithm based on dynamic programming.

Question 5

Give the recursion formula to define the value of $Sol[i, j]$ using $Sol[i - 1, j]$ and $Sol[i - 1, j - 1]$.

Question 6

Go back to the practice page of the TD to write in Python an algorithm that uses the optimal substructure explained in the previous question. Give the complexity of the algorithm.

Maximum flow – Minimum cost [advanced]

The advanced students can explore a second technique to compute an optimal solution of the ski rental problem.

The maximum flow of minimum cost problem is as follows. Given that :

- a graph oriented $G = (V, E)$;
- two source and sink vertices $sinV$, $tinV$;
- a capacity function $cap : E \rightarrow mathbbN$;
- a cost function $cost : E \rightarrow mathbbN$;

$\Rightarrow$ find a maximum flow f: $E \rightarrow \mathbb{N}$, such that $\sum_{e \in E} cost(e) * f(e)$ is minimum.

Question 7

Propose a model to solve the skis rental problem using a maximum flow minimum cost problem.

Indication: We are looking for a graph with unary capacities to force the flow to be 0 or 1.

Question 8

To code a Python algorithm to solve such a problem, go back to the practice page of the TD.