Algorithms SV worksheet 2

Bálint Molnár

August 29, 2025

1 Dynamic Programming

1. You are given n boxes, so that the weight of the i-th box is $w[i]$ and the height of the box is $h[i]$. You would like to build a tower by putting some boxes on top of each other. However, you can only put box A on top of box B if A has a smaller height and smaller weight. Design an algorithm to find the maximum height of such a tower.

2. You are given an $N \times M$ grid. The field in row i and column j contains $A[i, j] \geq 0$ golden coins, Some fields are forbidden: If the field in row i and column j is forbidden, $B[i, j] = 1$, otherwise $B[i, j] = 0$. Your goal is to go from the top left corner $(1, 1)$ to the bottom right (N, M), collecting as many coins as possible. In one step, you can move one step right or one step down. You are never allowed to move to forbidden fields.

   (a) Design an algorithm that finds the best route you can take (given N, M, A, B).

   (b) How would you change your algorithm if you are allowed to move one step up as well (you won't get extra coins if you visit a field twice. You are also not allowed to leave the grid)?

   (c) And if you can move to all four directions?

3. Recall that general trees can be defined in OCaml as

```
type 'a tree =
    | Node of 'a * 'a tree list
```

   Or equivalently as

```
class Tree<T> {
    private T value;
    private List<Tree<T>> children;
}
```

   in Java.

   Design an algorithm that finds the longest path in the tree. See example below:

2 Greedy Algorithms

1. N guests were invited. Each guest has indicated in advance the time intervals during which they will be present. The organizers want to capture all attendees in photographs during the event. Their goal is to take the minimum number of photos while ensuring that every guest appears in at least one photo.

   Design an algorithm that calculates the minimum number of photographs required and also determines the specific times when group photos should be taken.
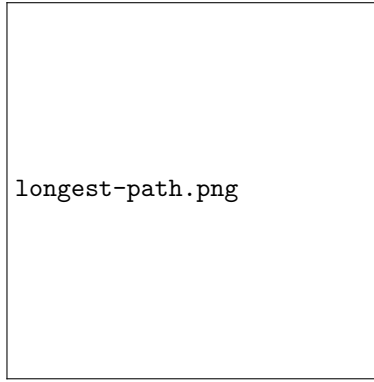
Figure 1: Longest path is labelled red

2. You are driving on a highway of length L kilometres, with a car with T units of fuel tank capacity. Initially, the tank is full, and every kilometre uses 1 unit of fuel. There are N petrol stations, and the i-th station is $A[i]$ kilometres away from your starting point.

   (a) Design an algorithm that, given L, T, N, A, finds the minimum number of times you need to refuel so you can get to the end of the highway, without running out of fuel. You can assume that all the inputs (L, T, N, and elements of A) are integers and A is sorted.

   (b) Now assume that the price of fuel is $P[i]$ in the i-th station for each i. Design an algorithm that finds the minimum cost of fuel you need to get to the end of the highway (there is no limit on how many times you stop for refuelling).