Reading material

We will introduce the paradigm *dynamic programming*. You should read KT Section 6.1, 6.2, and 6.4.

Exercises

1 [w] **Weighted interval scheduling** Solve the following weighted interval scheduling problem using both the recursive method with memoization and the iterative method. The intervals are given as triples (s_i, f_i, v_i): $S = \{(1,7,4),(10,12,2),(2,5,3),(8,11,4),(12,13,3),(3,9,5),(3,4,3),(4,6,3),(5,8,2),(4,13,6)\}$.

2 [w] **Knapsack** Solve the following knapsack problem by filling out the table below. The items are givens as pairs (w_i, v_i): $(5,7),(2,6),(3,3),(2,1)$. The capacity $W = 6$.

4								
3								
2								
1								
0								
$i \setminus w$	0	1	2	3	4	5	5	6

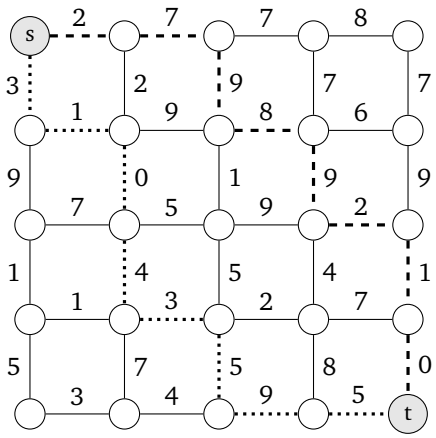
3 Independent set on a path Solve KT 6.1. In 6.1 (c) you should both write the recurrence and also write pseudocode for both a top-down and a bottom-up algorithm finding the value of an optimal solution. Also explain how to find the nodes contained in maximum weight independent set.

4 Job planning Solve KT 6.2.

5 Manhattan Tourists Your are helping the tourist guide company "Manhattan Tourists", that are arranging guided tours of the city. They want to find a walk between two points on the map that is both interesting and short. The map is a square grid graph. The square grid graph has n rows with n nodes in each row. Let node $v_{i,j}$ denote the jth node on row i. For $1 \leq i < n$ and for $1 \leq j \leq n$ node $v_{i,j}$ is connected to $v_{i+1,j}$. And for $1 \leq i \leq n$ and for $1 \leq j < n$ node $v_{i,j}$ is connected to $v_{i,j+1}$. The edges have non-negative edge weights that indicate how interesting that street is. See the graph below for an example of a 5×5 grid graph.

They want to find a short interesting walk from the upper left corner ($s = v_{1,1}$) to the lower right corner ($t = v_{n,n}$). More precisely, they want to find a path with the possible smallest number of edges, and among all paths with this number of edges they want the path with the maximum weight (the weight of a path is the sum of weights on the path).

All shortest paths have $2n-2$ edges and go from s to t by walking either down or right in each step. In the example below two possible shortest paths (of length 8) are indicated. The dashed path has weigth 38 and the dotted path has weight 30.



Let $W[i,j]$ be the maximal weight you can get when walking from s to $v_{i,j}$ walking either down or right in each step. Let $D[i,j]$ be the weight of the edge going down from $v_{i,j}$ and let $R[i,j]$ be the weight of the edge going right from $v_{i,j}$.

5.1 Fill out the table for the example above:

$W_{i,j}$	1	2	3	4	5
1					
2					
3					
4					
5					

5.2 Give a recurrence that computes $W[i,j]$. Argue that the recurrence is correct.

5.3 Write pseudocode for an algorithm *based on dynamic programming and your recurrence from Question 5.2* that computes the maximum weight a shortest path can have in the grid graph.

Analyze the space usage and running time of your algorithm in terms of n.

5.4 [†] Implement your algorithm on CodeJudge (see CodeJudge for input/ouput format).

6 Office switching Solve KT 6.4.