CSE 431/531: Algorithm Analysis and Design Fall 2021

Homework 4

Instructor: Shi Li **Deadline: 11/18/2021**

Your Name: _____ Your Student ID: _____

Problems	1	2	3	Total
Max. Score	25	25	30	80
Your Score				

Problem 1 In this problem, we consider the weighted interval scheduling problem, with an additional constraint that the number of intervals we choose is at most k.

Formally, we are given n intervals $(s_1, f_1], (s_2, f_2], \cdots, (s_n, t_n]$, with positive integer weights $w_1, w_2, \cdots, w_n$, and an integer $k \in [n]$. The output of the problem is a set $S \subseteq [n]$ with $|S| \leq k$ such that for every two distinct elements $i, j \in S$, the intervals $(s_i, f_i]$ and $(s_j, f_j]$ are disjoint. Our goal is to maximize $\sum_{i \in S} w_i$.

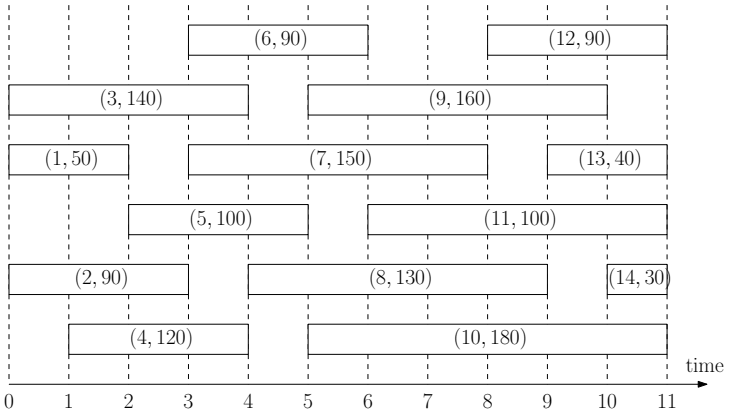

Figure 1: Weighted Job Scheduling Instance. Each rectangle denotes an interval. The first number on each rectangle is the index of the interval and the second number is its weight.

For example, consider the instance in Figure 1, with $k = 3$. Then the maximum weight we can obtain is 330, by choosing the set $\{2, 7, 12\}$ or $\{1, 5, 10\}$. If $k = 4$, then the maximum weight we can obtain is 340, with the set $\{1, 5, 9, 14\}$.

Design an $O(n \log n + nk)$-time dynamic programming algorithm to solve the problem. For simplicity, you only need to output the maximum weight that can be obtained. You need to give the definition of the cells and show how to compute the cells.

Problem 2 (From leetcode.com) There are n balloons in a row. Each balloon is painted with a positive integer number on it. You are asked to burst all the balloons.

If you burst the i-th remaining balloon in the row, you will get $nums[i-1] \times nums[i] \times nums[i+1]$ coins, where $nums[t]$ for any t is the number on the t-th remaining balloon in the row. If $i-1=0$ or $i+1$ is more than the number of balloons, then treat it as if there is a balloon with a 1 painted on it.

Return the maximum coins you can collect by bursting the balloons wisely. For example, if there are initially 4 balloons in the row with numbers 3,1,5,8 on them. Then the maximum coins you can get is 167. This is how the array of numbers change when you burst the balloons: $(3,1,5,8) \to (3,5,8) \to (3,8) \to (8) \to ()$. The coins you get is $3 \times 1 \times 5 + 3 \times 5 \times 8 + 1 \times 3 \times 8 + 1 \times 8 \times 1 = 167$.

Design an $O(n^3)$-time algorithm to solve the problem. For convenience, you only need to output the maximum number of coins you can get.

Problem 3 This problem asks for the maximum weighted independent set in a $2 \times n$ size grid. Formally, the set of vertices in the input graph G is $V = \{1,2\} \times \{1,2,3,\cdots,n\} = \big\{(r,c) : r \in \{1,2\}, c \in \{1,2,3,\cdots,n\}\big\}$. Two different vertices (r,c) and (r',c') in V are adjacent in G if and only if $|r-r'|+|c-c'|=1$. For every vertex $(r,c) \in V$, we are given the weight $w_{r,c} \geq 0$ of the vertex. The goal of the problem is to find an independent set of G with the maximum total weight. (Recall that $S \subseteq V$ is an independent set if there are no edges between any two vertices in S.) See Figure 2 for an example of an instance of the problem.

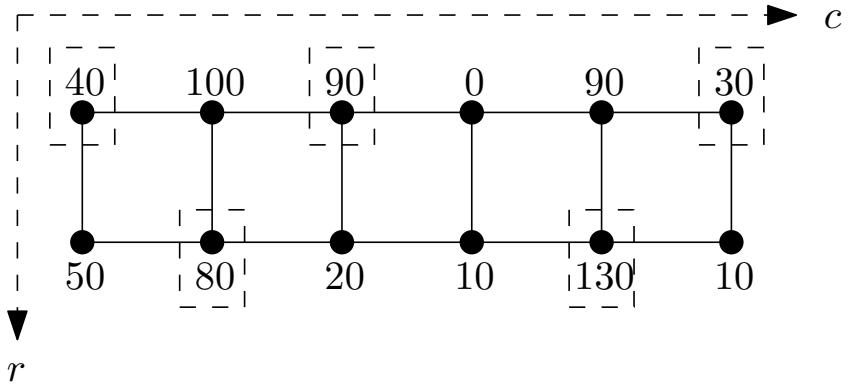

Figure 2: A maximum weighted independent set instance on a 2×6-grid. The weights of the vertices are given by the numbers. The vertices in rectangles form the maximum weighted independent set, with a total weight of 370.

Design an $O(n)$-time dynamic programming algorithm to solve the problem. For simplicity, you only need to output the *weight* of the maximum weighted independent set, not the actual set.

If you could not solve the above problem, you can try to solve the simpler problem when the grid size is $1 \times n$ instead of $2 \times n$ (that is, the input graph is a path on n verticies). If you solve the simpler case correctly, you will get 70% of the score.