You are allowed to discuss with others but are not allowed to use any references other than the course notes. Please list your collaborators for each question. You must write your own solutions.

There are totally 60 marks. The full mark is 50. This homework is counted 5% of the course grade.

---

1. **Programming Problem: Knapsack** (20 marks)

   You have N data files that you want to store, and your computer has W megabytes of storage. Each file i has size w_i megabytes and an importance score s_i.

   You want to find a subset of files to store such that the total file size does not exceed W while the sum of the importance scores is maximum.

   INPUT: The first line contains two integers N, W. Then N lines follow where the i-th lines contains two integers w_i and s_i. It is guaranteed that $1 \leq N, W \leq 2,000$ and for each $1 \leq i \leq N$, $0 \leq w_i, s_i \leq 10,000$.

   OUTPUT: The first line of the output should contain two integers S and K, the maximum possible importance score and the number of files you choose to store. The second line should contain K integers, the file indexes you choose to store. If there are more than one subset of files you can choose to store, you can output any one of them. You can also output the indexes of the files in any order.

   SAMPLE INPUT:
   ```
   5 10
   3 4
   4 6
   4 4
   5 8
   6 7
   ```
   SAMPLE OUTPUT:
   ```
   14 2
   4 2
   ```

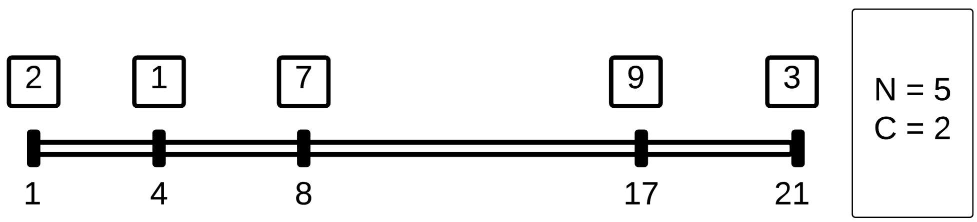
2. **Written Problem: Fill The Tank** (14 marks)

There are N cities named with numbers $1, \ldots, N$. These cities are located from left to right on the x-axis. City i locates at coordinates x_i and it is guaranteed that $x_i < x_{i+1}$ for $1 \leq i < N$. Let $M = x_N - x_1$. One day you want to travel from city 1 to city N. There are $N - 1$ one way roads, where the i-th road goes from city i to city $i + 1$, and the length of the road is $(x_{i+1} - x_i)$ kilometers long.

Your car travels 1 kilometer and consumes 1 liter of fuel per hour.

Each city i there is a fuel station which you can transfer s_i liters of fuel to your fuel tank for every C hours. Initially your fuel tank is empty with unlimited capacity. Of course, the constraint is that you cannot continue to travel if your tank is emptied somewhere strictly between two cities.

Design an algorithm to find the minimum time for you to travel from city 1 to city N. Your algorithm should return the full schedule of where to fuel the tank and by how much (see the following example). You can assume all input mentioned are positive integers. You will get full marks if the time complexity is $O(N)$ and the proofs are correct. You will get partial marks as long as the time complexity is polynomial in N and M and the proofs are correct.



Explanation: In the above example, we have $N = 5, C = 2$, $x = [1, 4, 8, 17, 21]$ and $s = [2, 1, 7, 9, 3]$. Then the optimal schedule to fill the tank would be:

Time	Curent city	Remaining fuel	Action
$t = 0$	City 1	0 L	Fill the tank in city 1 four times.
$t = 8$	City 1	8 L	Move to city 2, then move to city 3.
$t = 15$	City 3	1 L	Fill the tank in city 3 two times.
$t = 19$	City 3	15 L	Move to city 4, then move to city 5.
$t = 32$	City 5	2 L	Arrived!

Hence the minimum time travel from city 1 to city 5 is 32 hours.

3. **Written Problem: Thickest Paths** (12 marks)

Imagine that an online video provider would like to find paths of highest bandwidth to send its videos to the receivers. This can be modeled as a graph problem. We are given a directed graph $G = (V, E)$ where every directed edge e has a positive integer thickness b_e (representing the bandwidth of a link). Given a source vertex s and a receiver vertex t, let $\mathcal{P}_{s,t}$ be the set of all directed paths from s to t in G. We would like to find a path in $\mathcal{P}_{s,t}$ that maximizes the minimum thickness on the path, i.e.,

$$\max_{P \in \mathcal{P}_{s,t}} \min_{e \in P} b_e.$$

Design an efficient algorithm to find a thickest path from s to t for all $t \in V - s$. You will get full marks if the time complexity of the algorithm is $O((|V| + |E|) \log |V|)$ and the proofs are correct.

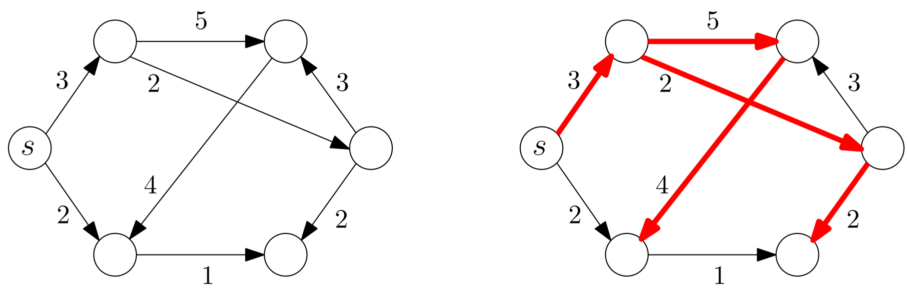


Figure 1: The highlighted edges on the right form the thickest paths from s to all other vertices.

4. **Written Problem: Maximum Interval Coloring** (14 marks)

   In class, we have studied the interval scheduling problem and the interval coloring problem. Here we consider a common generalization of these two problems. We are given n intervals $[s_i, f_i]$ where each s_i and f_i are positive integers with $s_i < f_i$, and a positive integer k. Our goal is to use k colors to color as many intervals as possible, so that each interval receives at most one color (could have no color on an interval) and no two overlapping intervals can receive the same color (two intervals $[s_i, f_i]$ and $[s_j, f_j]$ are overlapping if $[s_i, f_i] \cap [s_j, f_j] \neq \emptyset$). Notice that the interval scheduling problem is the special case when $k = 1$, and the interval coloring problem is the special case to find the minimum k so that all the intervals can be colored. You can think of this problem as using k rooms (k colors) to schedule as many activities (intervals) without time conflicts as possible.

   Design an efficient algorithm to find such a coloring. You will get full marks if the time complexity of the algorithm is $O(n \log n)$ and the proofs are correct. To implement the algorithm efficiently, you can use a balanced search tree data structure (such as AVL tree) and assume all the operations (add, delete, search) can be done in $O(\log |T|)$ time where $|T|$ is the size of the search tree.

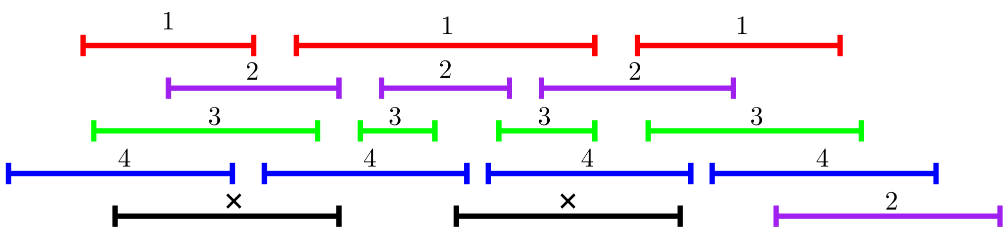


Figure 2: Coloring these intervals with four colors. The intervals with a cross are uncolored.