

## CoPA Questions

For problems 1-3, you should solve them using **CoPA**  
and document the **entire** process (questions and answers) in a .md file.

---

### 1. Question1

Professor W is planning a series of space flights for the National Space Center. Each space flight can perform a series of commercial experiments to generate profit. A set of selectable experiments is defined as  $E = \{E_1, E_2, \dots, E_m\}$ , and the instruments required for these experiments are defined as  $I = \{I_1, I_2, \dots, I_n\}$ . Each experiment  $E_j$  requires a subset of instruments  $I$   $R_j \subseteq I$ .

The cost of configuring instrument  $I_k$  is  $c_k$  dollars. The sponsor of experiment  $E_j$  agrees to pay  $p_j$  dollars for the experiment's results. Professor W's task is to design an efficient algorithm to determine which experiments to conduct and which instruments to configure in order to maximize the net profit of the space flight. Here, the net profit is defined as the total revenue generated from the experiments minus the total cost of configuring the instruments.

For the given experiments and instrument configuration data, write a program to find the experiment plan that achieves the maximum net profit.

Input format:

The first line contains two positive integers  $m$  and  $n$ , where  $m$  is the number of experiments and  $n$  is the number of instruments. The next  $m$  lines describe each experiment. The first number on each line is the sponsor's payment for the experiment, followed by the indices of the instruments required for the experiment. The last line contains  $n$  numbers, representing the configuration costs for each instrument.

Output format:

The first line contains the indices of the selected experiments. The second line contains the indices of the selected instruments. The third line contains the maximum net profit

### 2. Question2

Due to humanity's excessive consumption of natural resources, people have realized that by around the year 2300, Earth will no longer be habitable. As a contingency plan, new green areas have been established on the Moon to facilitate migration when necessary. Unexpectedly, in the winter of 2177, Earth's environment suffered a chain collapse for unknown reasons, forcing humanity to migrate to the Moon in the shortest possible time.

There are  $n$  space stations located between Earth and the Moon, and  $m$  public transport spaceships shuttling between them. Each space station can hold an unlimited number of people, but each spaceship has a limited capacity. The  $i$ -th spaceship can carry at most  $h_i$  people. Each spaceship periodically stops at a series of space stations, e.g.,  $(1, 3, 4)$  indicates that the spaceship will stop at stations 134134134... cyclically. Traveling from any space station to any other via a spaceship takes a fixed time of 1. People can only board or disembark at a space station (or the Moon or Earth) when a spaceship stops there.

Initially, all people are on Earth, and all spaceships are at their starting stations. Design an algorithm to find the transportation plan that moves everyone to the Moon as quickly as possible.

Input format:

The first line contains three space-separated integers representing the number of space stations  $n$ , the number of spaceships  $m$ , and the number of people on Earth  $k$ . From the second to the  $(m+1)$ -th lines, each line describes the details of a spaceship. The  $(i+1)$ -th line begins with an integer  $h_i$ , the capacity of the  $i$ -th spaceship. This is followed by an integer  $r_i$ , the number of space stations the spaceship stops at. The next  $r_i$  integers represent the sequence of station IDs  $S_{i,j}$  where the spaceship stops cyclically, with station IDs ranging from 1 to  $n$ , Earth represented by 0, and the Moon represented by  $-1$ .

Output format:

Output a single integer representing the minimum time required to transfer everyone to the Moon. If it is not possible, output 0.

### 3. Question3

Muxii is a snow mage. With a flick of his magic wand and a whispered incantation, snowflakes fall from the sky and accumulate into a thick layer of snow on the ground. Thanks to Muxii's incredible magical powers, he is entrusted by the gods to control the snow across the entire world.

One day, the gods assigned Muxii a task: he must create a snow layer of a given thickness on the ground. The ground has a length of  $n$  and each position  $i$  on the ground needs to have a snow thickness of  $a_i$ . Muxii can only increase the snow thickness over a contiguous subarray of the ground by 1 unit of snow at a time, and the maximum length of the subarray that can be modified in a single operation is  $m$ . Given the constraints, Muxii wants to know the minimum number of operations needed to achieve the required snow thickness.

The initial array is an array of zeros, i.e.,  $[0, 0, 0, \dots, 0]$  with length  $n$ .

Each operation consists of selecting a subarray of length  $\leq m$  and increasing the snow thickness by 1 unit over all positions in that subarray.

Given an array  $a$  of length  $n$ , where  $a_i$  represents the snow thickness required at position  $i$ , you need to answer  $q$  queries. For each query, given the indices  $l$  and  $r$ , you must determine the minimum number of operations required to modify a subarray of length  $r - l + 1$  (i.e., the subarray  $a[l], a[l+1], \dots, a[r]$ ) to match the values in  $a$ .

Input format:

The first line contains three integers  $n, m, q$ , where:  $n$  is the length of the array,  $m$  is the maximum length of the subarray in each operation,  $q$  is the number of queries. The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$ , where  $a_i$  represents the required snow thickness at position  $i$ . The next  $q$  lines contain two integers  $l$  and  $r$  for each query, representing the indices of the subarray in question.

Output format:

For each query, output a single integer, the minimum number of operations needed to match the subarray  $a[l], a[l+1], \dots, a[r]$  to the target snow thickness.