

✂ The Monster Hunt ✂

Programming Assignment 2

The last rays of the sun illuminate the flashy road ahead of you. It is a long dangerous road that passes through n distant cities. Once peaceful oases in the hostile wilderness, each city is now plagued by m nasty monsters. "If I were not so old and retired, I would go and kill them all!", you think, "And I would be rich as each monster guards a pile of gold coins. And I could even take home some mighty weapons of the monsters."

But the reality is that you have just a boring job at the information center at the beginning of the road. And thus you can only dream and look at the plan showing how much gold coins each monster is guarding, what weapon it is carrying and what type of weapon one needs to kill it. You also see a list showing all k weapon types that exist as well as the list of rules of the kingdom:

- *You are allowed to carry at most one weapon.*
- *You are allowed to kill at most one monster per city.*
- *You get a free weapon of any desired type at the information center.*

As it is getting darker, more and more people show up and approach you one after the other. "In total h of them this night.", you count. Each of them has the same type of tedious question:

"How much gold coins can I possibly earn from hunting monsters when following the road up to some city i and assuming that no one else is hunting?"

You yawn but then recall that the faster you answer, the larger will be the tip (grade) you get. And if you compute the answer in your head without using extra papers to sketch the computations (memory complexity is low), you even get an some extra tips!

Problem Definition

Input:

(At the end, you find the formal definition of the input format.)

- n, m, k, h (positive integers)
- $\text{attack}[0..n-1][0..m-1]$ (2D array with integers from 0 to $k-1$)
- $\text{weapon}[0..n-1][0..m-1]$ (2D array with integers from 0 to $k-1$)
- $\text{gold}[0..n-1][0..m-1]$ (2D array with nonnegative integers)
- $\text{city}[0..h-1]$ (array with integers from 0 to $n-1$)

Interpretation:

There are n cities numbered from 0 to $n-1$ that are visited in this order. Each city contains m monsters, numbered from 0 to $m-1$. In total, there are k different weapon types numbered from 0 to $k-1$.

Send Feedback

To kill the j -th monster of the i -th city, you need a weapon of type $\text{attack}[i][j]$. If you kill the monster, you obtain $\text{gold}[i][j]$ many gold coins and you can replace your current weapon with a weapon of type $\text{weapon}[i][j]$. You are allowed to kill at most one monster per city.

There are also h heroes. Each hero asks for the maximum amount of gold coins that one can gain after reaching the city $\text{city}[i]$ when the weapon in the beginning can be chosen freely (and assuming that no one else is killing monsters). Note that there might be more heroes than cities.

Output:

- $\text{answer}[0..h-1]$ where $\text{answer}[i]$ is the answer to the question of the i -th hero.

Example

Consider the example to the left. There are three cities ($n=3$), each having two monsters ($m=2$). There are three weapon types ($k=3$): sword (type 0), bow (type 1), ax (type 2).

The corresponding input is as follows:

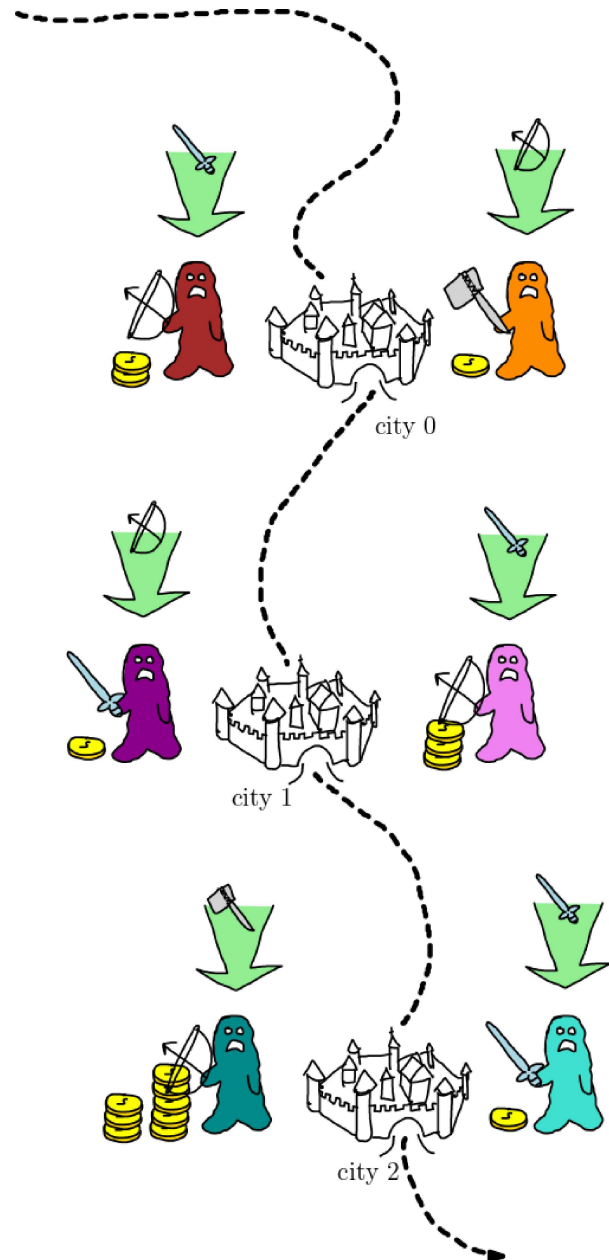
- $n=3, m=2, k=3, h=2$
- $\text{attack}=[[0, 1],[1, 0],[2, 0]]$
- $\text{weapon}=[[1, 2],[0, 1],[1, 0]]$
- $\text{gold}=[[2, 1],[1, 3],[8, 1]]$
- $\text{city}=[1,2]$

We have two heroes, the first one asks "1" ($=\text{city}[0]$), that is, how much gold one can earn at most after reaching city 1. The answer is "5" and is obtained as follows:

- Initially choose the sword.
- Kill the left monster of city 0. This gives you two coins of gold. Don't change the weapon.
- Next, kill the right monster of city 1. This gives you three more gold coins. Hence, you have five gold coins in total as claimed.

The second hero asks "2". Given that "5" is the answer for the question "1", we know that the answer for the question "2" is at least "5". By the following strategy, we see that the correct answer is even "9":

- Initially choose the bow.
- Kill the right monster of city 0. This gives you one gold coin. Change your weapon to the monster's weapon which is an ax



Send Feedback

- Skip city 1 as you cannot attack any monster with your current weapon.
- In city 2, kill the left monster. This gives you additional eight gold coins. In total, you have nine gold coins.

Summarized, the correct output is

- answer = [5, 9].

Details

You can assume that

- In Test 1: n, m, k, h are less than 5.
- In Test 2-4: n, m, k, h are not larger than 10.
- In Test 10, 11: n, m, k, h are not larger than 100.
- In Test 20, 30: n is not larger than 5000, m and k not larger than 32000 and at any point their product $n*m*k$ is bounded by roughly 50 000 000 000 (50 billion) .

Formally, the input of your program is given via the standard input and your output has to be written into the standard output. The format of the input and the output is described at the end of this page. However, you don't need to worry about it. The following code (one in Python and one in C++) automatically reads the input and writes the output. You can use it as a base of your program! (But you are free to write everything from scratch yourself.)

[Example Code in Python (Updated)] (material/base-updated.py) [Example Code in C++] (material/base.cpp)

Rules

- The **deadline is on Tuesday, 11 May** 10 minutes before the lecture (16:35 Polish time zone).
- Hint: use **dynamic programming** or **recursion with memoization**!
- Discussing the algorithm and the implementation with other people is not allowed.
- We will compare all solutions and take measures if we suspect plagiarism.
- Only C++ and Python are allowed.
- Using standard libraries is allowed—including numpy for Python (anything that works on the Szkopul webserver is allowed).
- Upload your program on the Szkopul webserver anytime before the deadline and how often you want. Shortly after each upload, you will see which tests you passed or failed and what your total score is.
- **Your last submission counts! Hence, please take care that your last submission is the one with the most points.**
- The first tests (Test 0a, 0b, 0c) are example tests for which we also provide the output (see below). These tests do not give you any points. Note that Test 0a is the example discussed above.
- The inputs of all the other tests are secret.

Send Feedback

- You get 0 points for a test if your program crashes, has the wrong output, is too slow or takes too much memory.
- To get the full score (100 points), your program has to pass all the tests within the respective time and memory limits. Note that this is only possible if the underlying algorithm has a sufficiently small time and memory complexity.
 - Tests 1-4 (50 points) can be easily passed if your algorithm computes the correct output even if it is far from being optimal with respect to time or memory complexity.
 - Test 10-11 (40 points) can be easily passed if you use dynamic programming or recursion with memoization.
 - Test 20 (5 points) can be only passed if your running time is optimal.
 - Test 30 (5 points) can be only passed if your memory complexity is low enough.

Recursion with memoization should give you at least 90 points, but it might not work for Tests 20 and 30. Also note that the test thresholds are chosen such that you have no disadvantage of using Python.

There are two time limits: If you are below the first time limit, you get full points. If you are between the two time limits, you get less points. If you are above the second time limit, you get 0 points. The second time limit is twice as large as the first one. In Szkopul, only the value of the second time limit is displayed.

There is also a generous memory limit that you have to pass.

- **Activity points:** You can earn up to two activity points by submitting a test on Moodle. The test should contain an input and a corresponding output in the format as specified below. It will be published to help other students.
- In the code above, you find in the comments some code by which you can read the input from a given file (input.txt) and write the output to another file (your-output.txt) and compare it to the intended output (output.txt). By this, you can use the test data submitted by other students to test your code.

Format

Input:

1. n m k h
2. (empty row)
3. n rows each consisting of m integers with values from $0..k-1$ (corresponds to the array "attack")
4. (empty row)
5. n rows each consisting of m integers with values from $0..k-1$ (corresponds to the array "weapon")
6. (empty row)
7. n rows each consisting of m integers with nonnegative values (corresponds to the array "gold")
8. (empty row)
9. h rows each containing one integer with values from $0..n-1$ (corresponds to the array "city")

Output:

1. h rows where the i -th row contains the answer to the i -th hero

Tests 0a, 0b and 0c have the following input and output. Note that Test 0a corresponds to the example above.

- [Input Test 0a] (material/input0a.txt) [Output Test 0a] (material/output0a.txt)
- [Input Test 0b] (material/input0b.txt) [Output Test 0b] (material/output0b.txt)
- [Input Test 0c] (material/input0c.txt) [Output Test 0c] (material/output0c.txt)

Send Feedback

Powered by OIOIOI (<https://www.github.com/sio2project/oioioi>), from the SIO2 Project (<http://sio2project.mimuw.edu.pl>).

[Send Feedback](#)