

CS325: Analysis of Algorithms, Fall 2017

Group Assignment 2*

Due: Tue, 10/24/17

Homework Policy:

1. Students should work on group assignments in groups of preferably three people. Each group submits to TEACH a zip file that includes their source code and their *typeset* report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded, and the codes submitted to teach will be tested.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy*: you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. You can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. More items might be added to this list. ☺

Vankin's Mile is an American solitaire game played on an $n \times n$ square grid. The player starts by placing a token on any square of the grid. Then on each turn, the player moves the token either one square to the right or one square down. The game ends when player moves the token off the edge of the board. Each square of the grid has a numerical value, which could be positive, negative, or zero. The player starts with a score of zero; whenever the token lands on a square, the player adds its value to his score. The object of the game is to score as many points as possible.

-1	7	-8	10	-5
-4	-9	8	-6	0
5	-2	-6	-6	7
-7	4	7	-3	-3
7	1	-6	4	-9

*The problem is from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on recursion is recommended.

For example, given the grid below, the player can score $8 - 6 + 7 - 3 + 4 = 10$ points by placing the initial token on the 8 in the second row, and then moving down, down, right, down, down. (This is not the best possible score for these values.)

In this assignment, you describe and analyze an efficient algorithm to compute the maximum possible score for a game of Vankin's Mile, given the $n \times n$ array of values as input.

Report (60%). In your report, include the description of your algorithm, and provide running time analysis. Algorithms should be explained in plain english. You can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.

Code (40%). Write a program to compute the maximum score for a given $n \times n$ board. Your program will be tested against several test cases, for correctness and efficiency. For each test case, the program will be automatically stopped after 20 seconds if it is not done in that time. In this case, the group will miss the points of that test case. Your program must be written in one of the following languages: Python, C++, or Java.

Input/Output Your program reads from one text file, "input.txt". The first line is one integer $1 \leq n \leq 1000$. The following n lines each is a row of the matrix. Each line is composed of n integers, each between -100 and 100 , separated by commas.

Your output must be written in "output.txt". The output is a single integer: the maximum possible score on the $n \times n$ board of the input.

Sample Input (1):

```
2
5,-2
-3,1
```

Sample Output (1):

```
4
```

Sample Input (2):

```
3
1,2,3
2,-10,-20
-20,20,-10
```

Sample Output (2):

```
20
```

Submission Each group submits to TEACH a zip file that includes their source code (*which must be just one file with name "vankin.cpp", "vankin.java", or "vankin.py"*) and their report in pdf format. This file can be submitted by any member of the group, but all names must be listed in the submitted report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under the door of my office before the midnight of the due day. The hard copy will be graded, and the submitted code to teach will be tested.

Your codes will be tested *automatically*. So, you need to carefully follow all formatting requirements mentioned above. To summarize:

- (1) Your source code file should be named “vankin.cpp”, “vankin.java”, or “vankin.py”.
- (2) It reads from files “input.txt” in the current folder.
- (3) It writes to the file “output.txt” in the current folder. It should write exactly one number into “output.txt”, with no extra symbol.

Test your code with the sample test files (<http://web.engr.oregonstate.edu/~nayyeria/CS325/Fall17/hws/test2.zip>) before submitting them, to make sure there is no formatting error.

Comments/hints:

- (1) This is a dynamic programming question. I recommend coming up with the recursive algorithm first, then memoize it, and (if you like) turn it into an iterative algorithm. For the report, it would be easier to see that you had the right idea if you describe and justify the recursion first.