IU-B, SOIC

Project (CSCI 503B)

Dynamic programming using matrix multiplication and Strassen

Erfan Sadeqi Azer

October 29, 2015

1 INTRODUCTION

For this project, you need to solve a programming problem on an online server called *Hacherrank*. So, first step is to open an account, if you do not have one yet. You can easily do so in Here.
You might try to solve some problems in "warmup" domain to get used to Hacherrank server. Starting with warmup problems is highly recommended, but they are not part of this project. The problem that you are going to solve for this project is Towers.
There are around 30 programming languages, allowed for this problem, including Python, C, and Java. The time limit for each language is different. Thus, choose the one that you are most familiar with.

2 ALGORITHM

After you read and understood the problem, you might first start with designing a dynamic programming algorithm, analyze time and space complexity. Then, try to improve the time complexity. This can be done by reformulating the solution using matrix notation, and then by a method called "Exponentiation by squaring." There is a "Discussions" tab in the problem statement page. Hackerrank users use it to discuss the problem as well as potential algorithms with each other and with the problem designer. You are very welcome to read those and ask or contribute to those discussions within Hackerrank's polices. Also, it is possible to find related discussions in the web. You are allowed to look for and use anything about theory part of the problem. For example, This page is where you can find a detailed explanation of the algorithm, that you are required to implement. However, you are not allowed to search for, read or use whole or part of any code that is written by anyone other than you.

3 IMPLEMENTATION

After you understood and analyzed your optimized algorithm, you can go ahead and implement it with your favorite language. After you checked the sample inputs, and your own custom inputs, it is possible to run your code on the server. You can do this by "Run Code" button at the bottom of the problem statement page. Then, you might submit your code and see if it is accepted by the online judge. If it is rejected, feel free to debug your code and submit again. Phase(I) of your project is completed once your submission is "accepted". Take a snapshot of the page, and keep it for your report.

4 STRASSEN

Strassen's algorithm is a method for multiplying two matrices with a lower time complexity, compared to naive algorithms[1]. It can be found in a lot of texts books, but its wiki page should be enough for this project. For Phase(II) of the project, you need to change the implementation of the matrix multiplication to this method, and then test to see which one is faster. To make sure there are no new bugs introduced with this method. Try to submit and get "accepted" status for your new code, as well.

5 REPORT

You need to write a short report, explaining your algorithm in your own words[2] with time and space analyses. Then explain your implementation. Mention a few main functions, and if you needed to do any implementation tricks to make it work. Include the snapshots of the both runs, with and without Strassen's algorithm. Also, compare the running time of these two methods and explain potential reasons. Make a zip file of your report along with your code, submit it through OnCourse system.

6 SOME MORE POINTS

- This project's implementation must be done individually. Each student should write their own code, character by character. Any form of copying from other students' codes or from an online resource, whether related or unrelated to this problem will be considered as plagiarism, and will not be tolerated. However, Discussing the algorithm and getting theoretical ideas from classmates or any other resource is allowed and encouraged.

- Although the algorithm is given and the implementation is straightforward, it is hard to estimate how long it will take to successfully finish the project. So, it is very important to start as soon as possible to have enough time to ask questions and fix potential bugs.

[1] Known as *ijk-method* or *ikj-method*, depending on the for loops order.

[2] This part should be in English. So, if you are giving a pseudo-code, it should not use any language-specific notations.

- Some languages have packages to do matrix computations, e.g., numpy for python. You are not allowed to use them whether hackerrank server recognizes or not. So, we are expecting to see implementations of matrix multiplication and exponentiation in your code. In case you have doubts if you are allowed to use a certain tool, ask beforehand.

- It is possible to solve this problem with different algorithms. For the sake of this project, you need to implement the algorithm, explained and referenced above. If you think you have a very creative algorithm, feel free to discuss it with your AI.

- Keep in mind that even though your code has been tested with an automatic judge, it will be read by your AI, too. So try to write a readable code. Make precise references in your report with names of functions or variables.