

BLG 335E – Analysis of Algorithms I

Fall2020 Homework 2

Deliver Date: 11.12.2020

Due Date: 25.12.2020

Important Notes:

- Please write your own codes, copying code parts from books, websites or any other sources including your friends is considered as plagiarism and results in minus points.
- Do not upload your codes any public platform (e.g. Github) until the deadline of homework passes.
- Use C++ language and do not forget to compile your codes on Linux using g++ command before sending them.
- You may use STL but do not use built-in functions for sorting.
- Do not forget to comment your code.
- Submit your source codes and report files on Ninova before the deadline, late submissions and submissions via e-mail will not be accepted.

Part 1. Implementation (50 points)

In this homework, you are expected to build a priority queue (PQ) for a taxi application simulation using the binary heap data structure. In the simulation scenario, a hotel desk clerk uses an application to see available taxis in the city with their distance to the hotel location. When a customer is leaving the hotel, the hotel desk clerk calls the nearest taxi for the customer. Your implementation should simulate the taxi application, i.e., showing the nearest taxis to the hotel using the provided coordinates of taxis.

You are given a dataset named "locations.txt". In the dataset, the locations of the taxis are provided as longitude and latitude. You will use the longitude and latitude information of a taxi to calculate its distance to the hotel by using the Euclidian distance. The location of the hotel is 33.40819 (longitude) and 39.19001 (latitude).

You need to implement a PQ using binary heap data structure that supports the following operations (30 points):

- A new taxi can become available in the city (in other words, a new distance can be added to the heap)

- Some taxis' distances to the hotel location can be updated (only decrease is allowed for this simulation)
- The hotel desk clerk may call the current nearest taxi (its information should be removed from the heap)

You need to develop a simulation for the taxi application with the following features (20 points):

- The closest taxi to the hotel location will be removed after every **100** operations.
- Each operation is an update with a probability of **p** or an addition of a new taxi with a probability of **1 - p**.
 - If an operation is an update, a random taxi's distance value will be decreased by **0,01**.
 - If an is an addition, new information will be read from the provided file, and the distance value of the new taxi will be added to the PQ.
- Simulation stops after **m** operations (new taxi addition, updating taxi distance, or calling nearest taxi).

Your program should run from the command line with the following format:

`./a.out m p`

The output must contain the following:

- The distance of the called taxis
- The number of taxi additions and distance updates
- Total running time in milliseconds

Part 2. Report (50 points)

1. (16 points) Please explain your implementation on the PQ operations and the simulation features with their theoretical running times. It is better you support your explanations with comments on your source code file.
2. (17 points) Demonstrate (using a graph or a table) of the effect of the **m** choice on the running time. You should run the simulation for different values of **m** between 1000 and 100000 for a constant **p** (0.2). Also, discuss the demonstration by answering following questions: Is the graph as what you have expected? Does it match to the theoretical running times?

3. (17 points) Demonstrate (using a graph or a table) the effect of the **p** choice on the running time. You should run the simulation for different values of p {0.1, 0.2, ..., 0.9} for a constant **m** (you should choose an appropriate value for **m**). Also, discuss the demonstration by answering following questions: Is the running time affected by **p**? Why do you think it is affected or not effected?