BLG336E - Analysis of Algorithms II **03.04.2018**
2017–2018 Spring, Project 2
Submission Deadline: 17.04.2018 23:59

Overview

In this project you are going to implement a suitable Divide-and-Conquer approach for the following problem.

Problem

In a three dimensional space (x, y, z), there are lots of identical balls which are hanging in free space. Suppose that balls start to enlarging at the same time and with same speed. Determine the initial distance between first pair of balls which are going to touch to each-other by using the data file you are given.

A) Implementation

1) Formalize this problem with a suitable divide-and-conquer approach.

- You may have a look at your course slides to find out which method you can use.

2) Use **Euclidean-distance** for the distance calculations.

3) Run your algorithm and analyze the results in terms of:

- total number of distance calculations
- the running time

4) Since your project will be tested with *another* data file, the input file should be chosen by a command line argument as in the given example.

Your program should compile and run using the following commands:

```
>g++ StudentID.cpp –o project2
>./project2 data1000.txt
The distance is 12.5624
Number of total distance calculations is 15632
```

Note that numbers in the example output is **imaginary**.

5) First line of the input file shows the total number of balls and other lines show the x, y and z locations of balls in the three dimensional space. Read the input file and store the values in to appropriate variables.

6) You will be given more than one input files. We want you to run your algorithm for all of them and compare their run times, show whether your complexity is correct. Put your `clock()` function at the right lines in your code and don't forget to specify time unit you used (ms, µs, etc.).

B) Report

1) Explain the master theorem with your own words briefly (max: 3 lines). What do a and b mean in divide-and-conquer approach? (max: 3 lines)

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

2) Present your problem formulation in detail.

3) How does your algorithms work?

- Write your pseudo-code.
- Write the time and space complexity of your algorithm on your pseudo-code.
- Write the recurrence relation of your algorithm. (T(base) and T(n)=?)

4) Analyze and explain the algorithm results in terms of:

- total number of distance calculations for each input file.
- the running time for each input file.

Note: If you have any questions, please feel free to contact Res. Asst. Emrullah GAZİOĞLU via e-mail (egazioglu@itu.edu.tr).

Policy:

● You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project.

● **Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions.**

Submission Instructions:

● Please submit your homework through Ninova e-Learning System.

● You must submit **all your source code in a single cpp file** and **a softcopy report (PDF).** You can define multiple classes in a single cpp file.

● All your code must be written in C++, and we must be able to compile and run on it on ITU's Linux Server (you can access it through SSH) using g++.

- For Windows users: If you wish, you can use WinSCP to upload your source code in to ITU SSH Server, and use PuTTY to compile and run your algorithm. If don't, **please make sure** that your code is able to compile and run on ITU's Linux Server.

● When you write your code, try to follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary.

● Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.

● You should be aware that the Ninova e-Learning System clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to Ninova, check to make sure that your project appears there.