Indian Institute of Technology Kharagpur

CS29003: Algorithms Laboratory, Spring 2022

Assignment 3: Divide-and-Conquer

2PM – 5PM 1st February, 2022

General Instructions (to be followed strictly)

Submit a single C/C++ source file.
Do not use global variables unless you are explicitly instructed so.
Do not use Standard Template Library (STL) of C++.
Use proper indentation in your code and include comments.
Name your file as `<roll_no>_a1.<extn>`
Write your name, roll number, and assignment number at the beginning of your program.

CADDIT is an streaming media service that offers a wide range of movies. It tries to match your preferences of movies with those of other users in order to recommend movies to you. Suppose you rank n movies. CADDIT looks up in its database for users with 'similar' interests. Let $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ and $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ be two rankings of the n movies. (These are just permutations of $1, 2, \ldots, n$). Then distance between the two rankings $d(\mathbf{r}, \mathbf{s})$ is defined as the number of pairs (i, j) (with $1 \le i < j \le n$) such that either $(r_i < r_j) \wedge (s_i > 2s_j)$ or $(r_i > 2r_j) \wedge (s_i < s_j)$. By renaming one of the rankings, say, $\mathbf{s}$ as $(1, 2, \ldots, n)$, the problem of computing $d'(\mathbf{r})$ defined as the number of pairs i, j such that $i < j$ and $r_i > 2r_j$.

(a) Write a function $dist_1$ that takes as input an array $\mathbf{r}$ and computes $d'(\mathbf{r})$ in $O(n^2)$ time, by looking at all pairs (i, j) and checking whether or not $r_i > 2r_j$.

(b) Write a function $dist_2$ implementing an $O(n \log n)$-time algorithm computing $d'(\mathbf{r})$.

In the *main()* function, read n and the rankings $\mathbf{r}$. Call the two functions and print the corresponding distances computed. Assume that $\mathbf{r}$ has the right form i.e., it is a permutation of $1, 2, \ldots, n$.

Sample Output 1

```
n = 10

Ranking: 4 9 1 7 3 10 6 2 8 5

Distance by Method 1: 8
Distance by Method 2: 8
```