Due: 11th of September 2017 at 11:59pm

COMP 2007 – Assignment 2

All submitted work must be done individually without consulting someone else's solutions in accordance with the University's Academic Dishonesty and Plagiarism policies.

IMPORTANT! Questions 1a–c, 2a(i–iii) and 2b(i–iii) should be submitted via Blackboard as pdf (no handwriting!). The implementation required for Questions 1d, 2a(iv) and 2b(iv) should be done in Ed, and submitted via Ed.

Questions

Assume we are about to go for dinner, and there are several restaurants to consider. To decide which restaurants to go to we might take the attributes of the restaurants into consideration. For example, the estimated price and the quality of the food. Of course, if every attribute of restaurant A is better than the attributes of restaurant B then restaurant A is always a better option than B. In this case we say that restaurant A is *superior* to restaurant B, and that B is *inferior* to A.

Figure 1 shows an example where we have five possible restaurants $p_1, \ldots, p_5$ and we only consider two attributes: price and quality. It is easy to see that there are restaurants that are not inferior to any other restaurant. The highlighted restaurants, p_1 and p_5, are not inferior to any other restaurants, however, restaurant p_1 is superior to p_2, p_3 and p_4 while p_5 is not superior to any other restaurant. We will use this observation to measure the importance of a restaurant. If a restaurant p_i is superior to k other restaurants then we say that its *importance* is k. Your task in this assignment is to compute the importance of every restaurant.

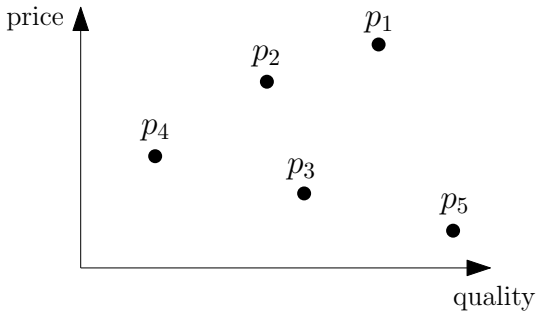

Figure 1: The two restaurants p_1 and p_5 are not inferior to any other restaurant but the importance of p_1 is 3 while the importance of p_5 is 0.

We now state the problem more formally. Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in d-dimensional space, where each point p_i is a d-tuple $p_i = (p_i(x_1), p_i(x_2), \ldots, p_i(x_d))$ representing its x_ℓ-coordinates, $1 \leq \ell \leq d$.

A point p_i is said to be *superior* to a point p_j if and only if $p_i(x_\ell) > p_j(x_\ell)$ for all ℓ, $1 \leq \ell \leq d$. In this case p_j is said to be *inferior* to p_i. If neither point is superior/inferior to the other then the points are *incomparable*. If a point p_i is superior to exactly k points in $P \setminus \{p_i\}$ then we say that its *importance* is k, denoted $imp(p_i) = k$.

Your task is to design algorithms in 1D and 2D that compute the importance of every point in P.

Note that the input test files follow the following name convention: The test cases for each of the three questions (1d, 2a(iv) and 2b(iv)) are `ns_ix.in` for the input files and `ns_ix.out` for the

output files, where `s` is the number of points $[10, 100, 200, 500, 1000]$ and `i` is an index in $[0, 1, 2]$. E.g. `n10_0.in` is the first input file on 10 points, `n10_1.in` is the second input file on 10 points and, `n10_2.in` is the third input file on 10 points. In total there are 15 test cases for each question.

1. **[20 points]** We will start with the simpler 1-dimensional case, that is, let $P = \{p_1, \ldots, p_n\}$ be a set of n points in 1-dimensional space (multiple points may have the same value). Your task is to design an $O(n \log n)$ time algorithm that computes the importance of each point in P.

   (a) Description of how your algorithm works ("in plain English"). [5 points]

   (b) Argue why your algorithm is correct. [4 points]

   (c) Prove an upper bound on the time complexity of your algorithm. [3 points]

   (d) Implement your algorithm (in Ed) and test it on the the given input instances. Each instance is given in a text file using the following format (where n is the number of points):

   $p_1(x_1)$
   $p_2(x_1)$
   ...
   $p_n(x_1)$

   You may assume that all the coordinates are given as non-negative integers. The output data file must use the following format:

   $\mathtt{imp}(p_{i_1})$
   $\mathtt{imp}(p_{i_2})$
   ...
   $\mathtt{imp}(p_{i_n})$

   where $p_{i_1}, p_{i_2}, \ldots, p_{i_n}$ are ordered according to increasing x_1-coordinates. **Important:** The points must be sorted otherwise the automatic tests will fail your submission. [8 points]

2. **[80 points]** Consider the same problem in 2D, that is, let $P = \{p_1, \ldots, p_n\}$ be a set of n points in 2-dimensional space, where each point p_i is a 2-tuple $p_i = (p_i(x_1), p_i(x_2))$. The task is to solve this problem using a divide-and-conquer approach. A divide and conquer algorithm works by recursively breaking down a problem into two (or more) sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then merged to give a solution to the original problem. For full marks on this question the algorithm is required to run in $O(n \log n)$ time.

   We will start with the merge step.

   (a) **[40 points]** Let $P_1 = \{p_1, \ldots, p_{n_1}\}$ and $P_2 = \{q_1, \ldots, q_{n_2}\}$ be two point sets in the plane that are separated by a vertical line (every point in P_1 lies to the left of all points in P_2). Also, the sets in P_1 and P_2 are ordered from bottom to top (in case of a tie the points are ordered with respect to increasing x_1-coordinates). Assume that the problem has been solved for P_1 and P_2, respectively. That is, the importance of each points in P_1 has been computed with respect to P_1 and the importance of each points in P_2 has been computed with respect to P_2. Your task is to combine the two solutions into one solution for $P = P_1 \cup P_2$. An example is shown in Fig. 2.

      i. Description of how your algorithm works ("in plain English"). [10 points]

      ii. Argue why your algorithm is correct. [10 points]

iii. Prove an upper bound on the time complexity of your algorithm. [5 points]

iv. Implement your algorithm (in Ed) and test it on the input instances. Each instance is given in a text file using the following format (where n is the total number of points, l is the number of points in P_1 and r is the number of points in P_2):

```
n
l
p1(x1) p1(x2) imp(p1)
p2(x1) p2(x2) imp(p2)
...
pl(x1) pl(x2) imp(pl)
r
q1(x1) q1(x2) imp(q1)
q2(x1) q2(x2) imp(q2)
...
qr(x1) qr(x2) imp(qr)
```

The output format should be in the following form for each of the output files:

```
imp(pi1)
imp(pi2)
...
imp(pin)
```

where $p_{i_1}, p_{i_2}, \ldots, p_{i_n}$ are ordered according to increasing x_2-coordinates (in case of ties, order with respect to increasing x_1-coordinates). **Important:** The points must be sorted otherwise the automatic tests will fail your submission. Note that this sorting step does not have to be described or analyzed. [15 points]

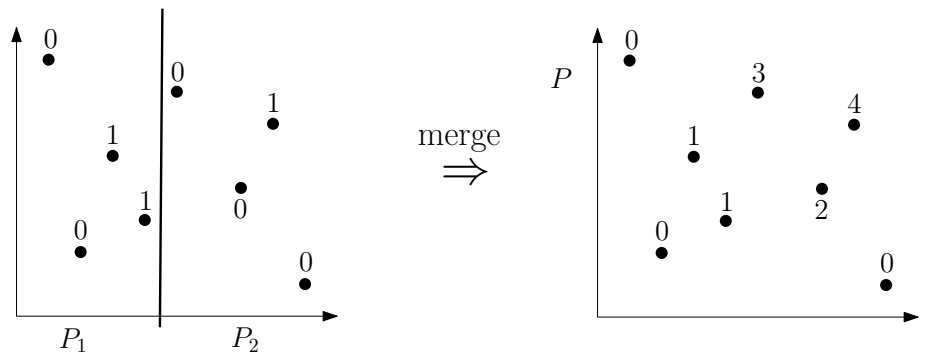

Figure 2: The two points set P_1 and P_2 are separated by a horizontal line. The merge step merges the input data into the importance of all the points in $P = P_1 \cup P_2$.

(b) **[40 points]** The final task is to use the merge step algorithm from (a) to construct a divide-and-conquer algorithm for the problem.

i. Description of how your algorithm works ("in plain English"). [5 points]

ii. Argue why your algorithm is correct. [10 points]

iii. Prove an upper bound on the time complexity of your (entire) algorithm. [10 points]

iv. Implement your algorithm (in Ed) and test it on the input instances. Each instance is given in a text file using the following format (where n is the number of points):

$\texttt{p}_1(\texttt{x}_1)\ \texttt{p}_1(\texttt{x}_2)$
$\texttt{p}_2(\texttt{x}_1)\ \texttt{p}_2(\texttt{x}_2)$
...
$\texttt{p}_\texttt{n}(\texttt{x}_1)\ \texttt{p}_\texttt{n}(\texttt{x}_2)$

The output format should be in the following form for each output file:

$\texttt{imp}(\texttt{p}_{\texttt{i}_1})$
$\texttt{imp}(\texttt{p}_{\texttt{i}_2})$
...
$\texttt{imp}(\texttt{p}_{\texttt{i}_\texttt{n}})$

where $p_{i_1}, p_{i_2}, \ldots, p_{i_n}$ are ordered according to increasing x_1-coordinates (in case of ties, order with respect to increasing x_2-coordinates). **Important:** The points must be sorted otherwise the automatic tests will fail your submission. Note that this sorting step does not have to be described or analyzed. [15 points]