This problem set has a total of 6 problems on 2 pages. Written solutions should be submitted to GradeScope before 9:00pm on Friday Oct 30.

Your solutions should be submitted according to the guidelines [https://www.cc.gatech.edu/~rpeng/CS4540_F20/ProblemSetGuidelines.pdf](https://www.cc.gatech.edu/~rpeng/CS4540_F20/ProblemSetGuidelines.pdf) In particular:

1. If you choose not to submit a typed write-up, please write neat and legibly.

2. No credit will be given to solutions obtained verbatim from the Internet or other sources, and uploaded codes will be ran through similarity checking software.

1. Implement a window manager in $O(\log^2 n)$ time: there is a collection of axis parallel rectangles, given in order from bottom to top. Every step, a click is given at some location, and the window clicked on gets brought to top. Given the initial $n$ rectangles and their ordering, followed by $n$ clicks, output the ids of the windows clicked on in $O(n \log^2 n)$ time.

   Autograder: [https://dmoj.ca/problem/cco05p6](https://dmoj.ca/problem/cco05p6) (Note: with some effort, it’s possible to sneak an $O(n^2)$ time solution by the data on there)

2. You are initially given an empty array $a$ of length $n$. There are $n$ operations to be performed as follows: Given an index $i$ corresponding to an empty position in the array, fill the position with a given integer $x$. That is set $a_i = x$, after which $a_i$ is no longer empty.

   After each of the $n$ operations, you are asked to print the total number of inversions in $a$. An inversion is defined as a pair $(i, j)$ with $1 \leq i < j \leq n$ such that $a_i$ and $a_j$ are non-empty with $a_i > a_j$. Give an $O(n \log^2 n)$ algorithm to solve this problem.

   Autograder: [https://dmoj.ca/problem/dmopc19c7p5](https://dmoj.ca/problem/dmopc19c7p5)

3. There is a checkerboard grid of $m \times n$ cells, all initially holding the value 0. Each cell is colored with one of two different colors: cells that share an edge have different colors. You are given $q$ queries of the following types:

   - **Update**: Set the value of cell $i, j$ to $v$.
   - **Query**: You are given opposite corners $(i_1, j_1)$ and $(i_2, j_2)$ of a rectangle in the grid, and are asked to return the sum of all integers in the grid with cell color equal to that of $(i_1, j_1)$ minus the sum of all integers in the grid with cell color different from that of $(i_1, j_1)$.
Give a solution to solve all queries that works in $O(mn + q \log n \log m)$ time and takes $O(mn)$ space.

Autograder: [https://dmoj.ca/problem/checker](https://dmoj.ca/problem/checker)

4. There are $n$ axis-parallel rectangles on the 2-D plane, given in order of height. That is, rectangle 1 is at the bottom, 2 is ‘higher’, but may not overlap with 1, and rectangle $n$ is the highest. Determine in $O(n \log n)$ time which rectangles have non-zero area visible from above.

Autograder: [https://dmoj.ca/problem/coci18c2p5](https://dmoj.ca/problem/coci18c2p5)

5. There is a $m \times m \times m$ grid. You are given $n$ (not necessarily distinct) coordinates $(i, j, k)$ indicating there is an item at the cell belonging to each of these coordinates. Give an $O(n \log^3 m)$ algorithm to count the number of unordered pairs of items such that the Manhattan distance between the cells that the items are stored in is at most $d$. The Manhattan distance between cells at $(i_1, j_1, k_1)$ and $(i_2, j_2, k_2)$ is $|i_1 - i_2| + |j_1 - j_2| + |k_1 - k - 2|$

Autograder: [https://dmoj.ca/problem/ioi07p5](https://dmoj.ca/problem/ioi07p5) (note: this also has the 1-D and 2-D cases, which we will ignore when grading)

6. There is an $n \times n$ grid that initially contains items of $m$ distinct types. The $i^{th}$ type has at most $t$ items, and you’re given their locations.

Now, you may select any $k \times k$ subgrid of cells. The value of such a selection is the number of item types that appear at least once in the region.

Additionally there are several traps located in certain cells. If your selection includes a trap, the value of the selection is 0 regardless of how many items it contains. Each cell may contain any number of items or traps. Before any queries are made, there are only empty cells and items, no traps have been placed. Now, you are given $q$ queries of the following types:

- **Update**: add a trip, $(i, j)$
- **Query**: If you could choose any $k \times k$ subgrid with equal probability, what is the probability the value of your selection is greater than $x$?

Give a $O(n^2 + mt^2 + q \log^2 n)$ time approach to answer all queries.

Autograder: [https://dmoj.ca/problem/cc015p6](https://dmoj.ca/problem/cc015p6) (Note: this set up has $t \leq 4$, so exponential in $t$ also works)