Change log:

- (version 1) Problem 3: Added $O(2^n)$ to complexity.
- (version 2) Problems 4 & 5: Added that no bounds on preprocessing the initial arrays are needed.

This problem set has a total of 6 problems on 3 pages. Written solutions should be submitted to GradeScope before 4:45pm on Wednesday Oct 7.

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. You are given an array of $n$ non-negative integers indexed from 0 to $n - 1$. There are $q$ queries to be processed. The two types of queries are:

   (a) Update: Change the value of element at index $i$ to a specified non-negative integer.
   (b) Query: Return the minimum integer between indices $i$ and $j$ of the array (both ends inclusive).

Give an $O((n + q) \log n)$ algorithm to process all queries.

Auto judge: [https://dmoj.ca/problem/segtree](https://dmoj.ca/problem/segtree)

2. You are given an array $a$ of $n$ positive integers with $1 \leq a_i \leq n$. There are $q$ queries to be processed. The 3 types of queries are:

   (a) Update: Change the value of element at index $i$ to a specified positive integer $x$. ($1 \leq x \leq n$)
   (b) Sum Query: Return the sum of integers between indices $i$ and $j$ of the array (both ends inclusive).
   (c) Compare Query: Return the number of integers in $a$ with value less than a specified integer $x$. ($1 \leq x \leq n$)
Give an $O((n + q) \log n)$ algorithm to process all queries.

Auto judge: [https://dmoj.ca/problem/ds1](https://dmoj.ca/problem/ds1)

3. There is a tournament with $2^n$ players. The $i^{th}$ player has skill level $a_i$. A game is played between pairs of two players, the player with the higher skill is the winner. It is guaranteed that all skill values are unique. In the first round of the tournament, Player 1 plays Player 2 in the first game, Player 3 plays Player 4 in the second game, etc. In the second round the winner of the 1st game plays the winner of the 2nd game, the winner of the 3rd game plays the winner of the 4th game, etc. Rounds are played until only one player remains - the overall winner. You are asked to answer $q$ queries about the tournament:

(a) Update: The skill level of the $i^{th}$ player is changed to new value $x$. Skill levels are guaranteed to be unique after this operation.

(b) Tournament Query: Return the index of the overall winner of the tournament.

(c) Round Query: Return the number of rounds won by a player $i$.

Give an $O(2^n + qn)$ algorithm to process all the queries.

Auto judge: [https://dmoj.ca/problem/cco13p2](https://dmoj.ca/problem/cco13p2)

4. Show that the following two operations can be maintained on an array of $n$ integers in $O(\log n)$ time per operation:

(a) Change the value of the $i^{th}$ entry, $v[i]$ to $x$.

(b) For $l$ and $r$ given in input, compute the bitwise and of entries in the range $[l \ldots r]$, that is, $v[l] \& v[l + 1] \& \ldots \& v[r]$.

You can use any amount of preprocessing for the initial array.

Auto judge: [https://dmoj.ca/problem/ichb2017p3](https://dmoj.ca/problem/ichb2017p3)

**Note:** this is different than sum because the bitwise and cannot be deduced from the results of $[1 \ldots l]$ and $[1 \ldots r]$. The lecture notes, and end of slides, contain some discussions of how to return the ‘merge’ of ranges, instead of prefixes.

5. For some modulus $p$ (that fits in a single integer), maintain two arrays of integers, $x[1 \ldots n]$ and $y[1 \ldots n]$ under modifications to $x[i]$ or $y[i]$, and return, after each update, the value

$$\max_i y[i] \cdot \prod_{1 \leq j \leq i} x[i] \pmod{p}$$

in $O(\log n)$ time. Note that extra steps are needed to account for the fact that products of $x[i]$ may exceed the sizes of integers.
You can use any amount of preprocessing for the initial array.

Auto judge: [https://dmoj.ca/problem/ioi15p4](https://dmoj.ca/problem/ioi15p4) (WARNING: Borat-esque flavor text, highly suggest looking at the above formal description first.)

6. Given $n$ tasks, each takes 1 unit of time to complete, but has deadline $d[i]$ In $O(n \log n)$ time, find the minimum total number of swaps of adjacent elements so that all tasks are completely before their deadlines, or report no such arrangement is possible.

Auto judge: [https://dmoj.ca/problem/ccp20p2](https://dmoj.ca/problem/ccp20p2)