CS345 : Algorithms II
Semester I, 2020-21, CSE, IIT Kanpur

Assignment 2

Deadline : 11:55 PM, 24 September 2020.

Most Important guidelines

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the net or from a notebook or from other fellow students. Remember - **Before cheating the instructor, you are cheating yourself**. The onus of learning from a course lies first on you. So act wisely while working on this assignment.
- **Grading policy:**
  Only the **first 25** submissions that claim to solve the difficult problem will be graded out of 100 marks - any subsequent submission (31st onwards) will be graded out of 80 marks only.
- Refrain from collaborating with the students of other groups. If any evidence is found that confirms copying, the penalty will be very harsh. Refer to the website at the link: https://cse.iitk.ac.in/pages/AntiCheatingPolicy.html regarding the departmental policy on cheating.

General guidelines

1. There are three problems in this assignment: Difficult, Moderate, and Easy. The difficult one carries 100 marks, the moderate one carries 80 marks, and the easy one carries 60 marks. Attempt **only** one of them.

2. You are strongly discouraged to submit the scanned copy of a handwritten solution. Instead, you should prepare your answer using any text processing software (LaTex, Microsoft word, ...). The final submission should be a single pdf file.

3. You need to justify any claim that you make during the analysis of the algorithm. But you must be formal, concise, and precise. You may use the results proved in the class. But, if you wish to use any homework problem in your solution, you must provide its solution as well.

4. If you are asked to design an algorithm, you may state the algorithm either in plain English or a pseudocode. But it must be formal, complete, unambiguous, and easy to read. You must not submit any code (in C++ or C, python, ...).

5. **Naming the file**:
   The submission file has to be given a name that reflects the information about the assignment number, version attempted (difficult/moderate/esay), and the roll numbers of the 2 students of the group. For example, you should name the file as **D_2_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the difficult version of the 2nd assignment. In a similar manner, the name should be **M_2_Rollnumber1_Rollnumber2.pdf** and **E_2_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the moderate problem and the easy problem respectively of the 2nd assignment.

6. **Only one** student of a group has to upload the final submission. Be careful during the submission of an assignment. Once submitted, it can not be re-submitted.

7. Deadline is strict. Make sure you upload the assignment well in time to avoid last minute rush.

8. Contact TA at the email address: ssnair@iitk.ac.in for all queries related to the submission of this assignment. Avoid sending any such queries to the instructor.

Difficult

Dynamic Sequence with Rotation operation

Design and analyse a data structure for a dynamic sequence S that supports the following operations.

1. Insert(S, i, x): Insert a new element with value x at ith place in the sequence S.
2. Delete(S, j): Delete the element present at jth place in the sequence S.
3. Report(S, i): Report ith element of the sequence S.
4. Rotate(S, i, j): Rotate the sequence S starting from ith element to jth element. In other words, the ith element swaps its position with that of the jth element, $(i+1)$th element swaps its position with $(j-1)$th position, and so on.

You must ensure that each of the 4 operations mentioned above is executed in the worst case $O(\log n)$ time by your data structure, where n is the length of the sequence at the time of the operation.

Moderate

Dynamic Sequence with Min and Add together

Design a data structure for a dynamic sequence S of numbers that supports the following operations.

1. Insert(S, i, c): Insert a number with value c at ith place in the sequence S.
2. Delete(S, j): Delete the number present at jth place in the sequence S.
3. Report(S, i): Report the number present at ith place in the sequence S.
4. Min(S, i, j): Report the smallest number among all the numbers at places $i, i+1, \ldots, j$ in the sequence S.
5. Add(S, i, j, x): Add x to each number at places $i, i+1, \ldots, j$ in the sequence S.

You must ensure that each of the 5 operations mentioned above is executed in the worst case $O(\log n)$ time by your data structure, where n is the length of the sequence at the time of the operation.

Easy

Dynamic Sequence on Bits

Design a data structure for a dynamic sequence S of bits that supports the following operations.

1. Insert(S, i, b): Insert a bit with value b at ith place in the sequence S.
2. Delete(S, j): Delete the bit present at jth place in the sequence S.
3. Report(S, i): Report the bit present at ith place in the sequence S.
4. Flip(S, i, j): For each k such that $i \leq k \leq j$, flip kth bit of the sequence S. That is, if the kth bit was 0 previously, it becomes 1 after this operation; it the kth bit was 1 previously, it becomes 0 after this operation.

You must ensure that each of the 4 operations mentioned above is executed in the worst case $O(\log n)$ time by your data structure, where n is the length of the sequence at the time of the operation.

An important note for all the problems of this assignment:
A solution of each of these problems should clearly provide the following details:

1. The fields and their description at each node of the augmented BST.
2. A neat and well commented pseudocode of each procedure.

There is no need to provide any analysis of the time complexity for each operation if you are submitting the solution of Moderate problem or Easy problem. But, for the Difficult problem, you must also show formally how your implementation achieves $O(\log n)$ time for the Rotate operation.