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

Assignment I

Deadline : 11:55 PM, 17 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:**
  For the current assignment, the maximum marks of each problem is independent of the time of submission of the solution. But it might not remain so in future as follows.

  Go through the difficult problem of this assignment. Based on the past experience, the correct submission for this problem should be less than 25%. After grading, if we find that there are much larger percentage of groups submitting the correct solution for this problem, we shall permanently introduce the following grading policy for all the future assignments:

  "Only the first 30 submissions of the difficult version will be graded out of 100 marks. Any subsequent submission (31st onwards) will be graded out of 70 or 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 50 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.

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_1_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the difficult version of the 1st assignment. In a similar manner, the name should be **M_1_Rollnumber1_Rollnumber2.pdf** and **E_1_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the moderate problem and the easy problem respectively of the 1st 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: shreyasa@iitk.ac.in for all queries related to the submission of the assignment. Avoid sending any such queries to the instructor.

Difficult

Non-dominated points

Recall the problem of non-dominated problem discussed in the second lecture of this course. We discussed two algorithms for this problem. The first algorithm takes $O(nh)$ time, where h is the number of non-dominated points in the given set P. The second algorithm, which was based on the divide and conquer paradigm, takes $O(n \log n)$ time. As a part of this assignment, you have to design an $O(n \log h)$ time algorithm for non-dominated points. Interestingly, you have to use the insight from the first algorithm to just *slightly* modify the second algorithm so that its running time is improved to $O(n \log h)$. You must describe the algorithm and also provide the complete details of the analysis of its running time.
Remark: Note that $O(n \log h)$ is superior to $O(n \log n)$ in those cases where the number of non-dominated points are very few. In fact, it can be shown that if n points are selected randomly uniformly from a unit square, then the expected(average) number of non-dominated points is just $O(\log n)$.

Moderate

Convex Hull

In Lecture 2, we discussed an $O(n \log^2 n)$ time algorithm to compute the convex hull of a given set of n points in a plane. If we can improve the time complexity of the Conquer step of this algorithm to linear, this will result in an $O(n \log n)$ time algorithm for convex hull. You have to modify the current Conquer step so that it takes at most linear time. You must provide a complete analysis of the modified Conquer step as well.

Easy

Counting Double-Inversions

You are given an array A storing n numbers. A pair (i, j) with $0 \leq i < j \leq n-1$ is said to be a double-inversion if $A[i] > 2A[j]$. Design and analyze an $O(n \log n)$ time algorithm based on divide and conquer paradigm to compute the total number of double-inversions in A.