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

Assignment 3

Deadline : 11:55 PM, 1st October 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:**
  There will be no penalty for submission based on the time of submission as long as the submission is before the deadline.
- 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 75 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. 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_i_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the difficult problem of the ith assignment. In a similar manner, the name should be **M_i_Rollnumber1_Rollnumber2.pdf** and **E_i_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the moderate problem and the easy problem respectively of the ith 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: **Kbhanja@cse.iitk.ac.in** for all queries related to the submission of this assignment. Avoid sending any such queries to the instructor.

Difficult

A hierarchical metric

Suppose we are given a set of points $P = \{p_1, p_2, \cdots, p_n\}$, together with a distance function d on the set P; d is simply a function on pairs of points in P with the properties that $d(p_i, p_j) = d(p_j, p_i) > 0$ iff $i \neq j$, and that $d(p_i, p_i) = 0$ for each i.

We define a *hierarchical metric* on P to be any distance function τ that can be constructed as follows: We build a rooted tree T with n leaves, and we associate with each node v of T (both leaves and internal nodes) a *height* $h(v)$. These heights must satisfy the properties that (1) $h(v) = 0$ if v is a leaf, and (2) If x is parent of y in T, then $h(x) \geq h(y)$. We place each point in P at a distinct leaf in T.

Now, for any pair of points p_i and p_j, their distance $\tau(p_i, p_j)$ is defined as follows: We determine the lowest common ancestor v in T of the leaves containing p_i and p_j, and define $\tau(p_i, p_j) = h(v)$.

We say that a *hierarchical metric* τ is *consistent* with our distance function d if, for all pairs i, j, we have $\tau(p_i, p_j) \leq d(p_i, p_j)$.

Give a polynomial-time algorithm that takes the distance function d and produces a hierarchical metric τ with the following properties.

1. τ is consistent with d, and
2. If τ' is any other hierarchical metric consistent with d, then $\tau'(p_i, p_j) \leq \tau(p_i, p_j)$ for each pair of points p_i and p_j.

You must provide a proof of correctness for the algorithm. There is no need to analyse the time complexity of the algorithm as long as it is bounded by a polynomial of the input size.

Moderate

An interval overlapping problem

There is a set S of n intervals. Each interval is specified by its start time and finish time. Our aim is to compute the smallest subset R of intervals such that for every other interval $I \in S \backslash R$, there is at least one interval in the set R that overlaps with it. Design an efficient algorithm for this problem. You must provide a proof correctness of the algorithm. You may assume for simplicity that no two intervals have the same start time or finish time.

Easy

An interval scheduling problem

There is a set J of n jobs. Each job has a start time and finish time. There is a single processor and it can execute only one job at a time. Design an algorithm to compute the largest subset $R \subseteq S$ of jobs that can be executed on the processor. It can be observed that no two jobs in R are allowed to overlap. You may assume for simplicity that no two jobs have same start or end times. You must provide a proof of correctness for the algorithm.

An important note for all the problems of this assignment:
As you may notice that each problem in this assignment has a greedy algorithm. Though you are free to design the algorithm and establish its correctness in any way you wish, you are strongly encouraged to pursue the generic technique discussed in the class. Consider this assignment as an opportunity to fully learn and internalize this technique.