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

Assignment 5

Deadline : 11:55 PM, 30th 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 two problems in this assignment: Difficult and Easy. The difficult one carries 100 marks, the easy one carries 75 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 **E_i_Rollnumber1_Rollnumber2.pdf** if you are submitting the solution for the easy problem of the ith assignment.
6. **Each student of a group** has to upload the submission file separately unlike in the past when only one submission file per group had to be uploaded. 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: **ssaxena@iitk.ac.in** for all queries related to the submission of this assignment. Avoid sending any such queries to the instructor.

Difficult

1 An adventurous drive in Thar desert

Your friend X has recently purchased a **THUNDERBIRD** motorcycle from Royal Enfield in India. He is very excited and wants to drive it from some town s to another town d in Thar desert (think of Sahara desert if you are more adventurous). He is provided with the complete road map of the desert - the roads, their lengths and junctions (where two or more roads meet). The mobike has a very natural limitation - it can drive for c kilometers with full fuel tank. Since the destination is very far, X must plan his route so that he can refill the fuel tank along the way. Note that the shortest route may not necessarily be the feasible route. You friend is bit confused and scared - what if he gets lost in Thar desert due to improper planning.

Your friend is very proud of you since you are a student of IITK. He also knows that you have done a course on algorithms. He approaches you with full faith that you will help him find the *shortest feasible* route from s to d if it exists. Model the problem in terms of a weighted, undirected graph in which roads are edges, junctions are vertices, and some junctions have fuel stations, and design an efficient algorithm to find *shortest feasible* route from s to d and inform if no route is feasible. Assume that both s and d are also at junctions and the source s has a fuel station. Your algorithm should take time of the order of $O(mn \log n)$ where m is the number of edges and n is the number of vertices. You should not exploit the planarity of the input graph while designing your algorithm. In other words, your algorithm should work for any arbitrary undirected graph with positive edge weights and under the fuel constraint mentioned above. Also note that the shortest feasible route need not be a simple path. For example, consider the following situation : There are five towns : s, a, b, e, d where fuel stations are available at s, b, e only. The connecting roads are : (s, a) of length 200 km, (s, d) of length 300 km, (a, b) of length 40 km, (a, d) of length 150 km, (b, e) of length 200 km, (e, d) of length 200 km; and motor bike can travel 250 km with full fuel tank. For this road network, shortest feasible route from s to d is is $s \to a \to b \to a \to d$, and has length 430 km.

Easy

Searching for a special shortest path

Given a directed graph on n vertices and m edges, where each edge has been assigned a positive weight. There are two parameters that characterize *lightness* of a path P: the first parameter is its *length* $\ell(P)$ which is defined as the number of edges of P. The second parameter is its *weight* $w(P)$ which is defined as the sum of weights of all its edges. A path P_1 is said to be *lighter* than path P_2 if either $\ell(P_1) < \ell(P_2)$ or $\ell(P_1) = \ell(P_2)$ and $w(P_1) < w(P_2)$. A path P from u to v is said to be the lightest path from u to v if no other path from u to v is lighter than P. Design an $O(m + n)$ time algorithm that for a source s and a destination d, compute the lightest path from s to d.