CS345 : Algorithms II
Semester I, 2016-17, CSE, IIT Kanpur

Assignment 5

Deadline : 6:00 PM, Friday, 27 October 2016

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. Before cheating the instructor, you are cheating yourself. The onus of learning from a course lies first on you and then on the quality of teaching of the instructor. So act wisely while working on this assignment

- There are two exercises in this assignments. Each exercise has two problems- one *easy* and one *difficult*. Submit exactly one problem per exercise. It will be better if a student submits a correct solution of an easy problem that he/she arrived on his/her own instead of a solution of the difficult problem obtained by hints and help from a friend. Do not try to be so greedy :-).

1

Attempt exactly one of the following two problems.

1.1 Dynamic Reachability (*marks = 50*)

Let G be a directed graph on n vertices with no edges in the beginning. Let s be a designated source vertex. We receive a sequence of m edge insertions. Our aim is to maintain a Boolean array R with the following property after each edge insertion.

$R[j]$ = true if and only if there is a path from s to j consisting of edges present in the graph.

In the beginning $R[s]$ =true, and $R[i]$ =false for each $i \neq s$. Upon insertion of an edge (i, j), we invoke Procedure Update-R(i, j) to update R. This is a recursive procedure sketched below. Fill in the blanks appropriately.

Procedure `Update-R`(i, j)

---

if ....*and*................ **then**

  ....$\leftarrow$ true;

  foreach *neighbor q of* .... **do**

    ....;

Though the worst case time of handling an edge insertion may be quite large, the total time for processing m insertions is $O(m)$. To show this, do the following tasks.

1. State a potential function such that the amortized cost of handling an edge insertion is $O(1)$.
2. Express the actual cost of handling an edge insertion formally and precisely.
3. Show here using the potential function defined above that the amortized cost of handling an edge insertion is indeed $O(1)$.

1.2 Queue using 2 stacks (*marks = 25*)

This is a problem from the practice sheet. You have to use two stacks to implement a queue in such a manner that the amortized cost of each operation is $O(1)$.

2 Fibonacci heap

Attempt exactly one of the following two problems.

2.1 Tinkering with Fibonacci heap (*marks = 50*)

In the Fibonacci heap discussed in the class, as soon as a marked node v loses its second child, the subtree rooted at v is cut from its parent and added to the root list. What if the subtree rooted at a marked node v is cut from its parent and added to the root list only when it loses its 3rd child ? Will all the bounds still hold on the amortized time complexity of various operations on Fibonacci heap ? Give rigorous mathematical arguments to support your claim.

2.2 Core property of Fibonacci heap (*marks = 25*)

This problem is directly from the lectures. Let v be any node in Fibonacci heap. Show that the degree of v is $O(\log(s(v)))$, where $s(v)$ is the size of the subtree rooted at v.