CS6033 Homework Assignment 2[*]

Due September 17th at 5:00 p.m.
Turn in this assignment as a PDF file on NYU classes
No late assignments accepted[†]

1. (15 points) For the following array

6	77	11	21	60	15	-2	16	91	90

   (a) create a max heap using the algorithm we discussed in class (BUILD-MAX-HEAP)

   (b) remove the largest item from the max heap you created in 1(a), using the HEAP-EXTRACT-MAX function from the book. Show the array after you have removed the largest item

   (c) using the algorithm from the book, MAX-HEAP-INSERT, insert 91 into the heap that resulted from question 1(b). Show the array after you have inserted the item.

2. (15 points) Prove the correctness of `HEAP-DECREASE-KEY` using a loop invariant.

3. (10 points) For the following algorithm:

   (a) (5 points) Write the recurrence formula for the following function.[1]

   (b) (5 points) Draw the recursion tree for the following function and show the running time for each level.

```
MINIMUM($A, l, r$)
1)  if ($r - l == 0$)
2)      return $A[r]$
3)
4)  $lmin$ = MINIMUM($A, l, \lfloor (l+r)/2 \rfloor$)
5)  $rmin$ = MINIMUM($A, \lfloor (l+r)/2 \rfloor + 1, r$)
6)  PRINT($rmin, lmin$)
7)  if $rmin < lmin$
8)      return $lmin$
9)  else
10)     return $rmin$
```

[*]Many of these questions came from outside sources.
[†]If the class has started, your homework will be late and will not be accepted for credit.
[1]You do not need to solve the recurrence formula.

4. (25 points) Consider a *quad* max-heap. A *quad* max-heap is like a binary max-heap, but instead of 2 children, nodes have 4 children.

   (a) How would you represent a *quad* max-heap in a array?

   (b) What is the height of a *quad* max-heap of n elements in terms of n and 4?

   (c) Give an efficient implementation of `HEAP-EXTRACT-MAX`. Analyze its running time in terms of 4 and n.

   (d) Give an efficient implementation of `MAX-HEAP-INSERT`. Analyze its running time in terms of 4 and n.

   (e) Give an efficient implementation of `HEAP-INCREASE-KEY`(A, i, k). Analyze its running time in terms of 4 and n.

5. (10 points) In your last MMORPG you *rocked*.

   The weekend is about to arrive and it is time to find teammates again. The problem is, the word has leaked and now everyone (perhaps on the entire planet) wants to join your team.

   With so many people, you have no idea how you will determine how many people are on your team and who they are.

   You need a faster algorithm than the last time.

   Repeat homework 1, question 9 - but this time your algorithm must return a sorted list of players (where no player is listed twice) in time $o(n^2)$. You may call any algorithm as a subroutine that was presented in lecture 2. State what the running time is of your algorithm in big-Oh, Theta, and Omega notation using the tightest bounds possible[2].

6. (10 points) Oh no! The bug that prevents the MMO from scaling the difficulty of the last boss has been fixed.

   Each player now has an attribute *level*. For the upcoming battle you will choose the top k players with highest level.

   Of course, word keeps getting around especially when people hear who has joined your team, so you will perform the steps in the following manner.

   - You will invite the best of all the players who want to join your team
   - After inviting that player you will see if any new players asked to join your team (after seeing who you already have on your team - more people will want to join!). Update accordingly
   - Ask the next best player to join
   - Afterwards again checking for new people who want to join and updating accordingly
   - Keep following these steps till you have k players

   You may use $O(k)$ space, $O(\log k)$ to invite the next player, and as new potential candidate arrives you can only spend $O(\log k)$ time per player.

   Design an algorithm. The fate of your gaming career depends on getting this right!

   Justify the run time of your algorithm and the justify the space requirement of your algorithm.

[2]In this course, you must always provide the tightest bounds, even if a looser bound is valid.

7. (10 points) The big space galactic grand tour tournament is about to begin, and ships are arriving from all over the galaxy. There are k portals for where ships can land on your ship. Each is near one of the k wormholes near your spaceship. Ships arrive from a wormhole line up outside the closest portal to that wormhole.[3]

   It will prevent some future wars if the people who are allowed in the main tournament hall located on the second deck is determined not by how many weapons they are carrying, but based on the time they exited the wormhole.

   Your job is to keep the lines moving and assign a number to each person based on the time they entered through a wormhole as efficiently as possible. You must choose the next person in $O(\log k)$ time.[4]

8. (15 points) Show that there are at most $\lceil \frac{n}{2^{h+1}} \rceil$ nodes of height h in any n-element heap.

9. OPTIONAL (10 bonus points)

   The intergalactic vote for confidence (a number from 0 to 100) in the supreme chancellor is tomorrow! If over 50% of the galaxies are not confident in the commander (i.e. have a confidence score of 50 or higher) he will be cast out the airlock!

   As the newspaper reporter for the galaxy times, you want to keep track and analyze of the scores as they come from the far flung galaxies.

   How can you return the median in $O(1)$ time and enter a new score in $O(\log n)$ time where n is the number of galaxies whose scores have already been entered.

   Reminder: The median of $10, 19, 21$ is 19. The median of $10, 30, 90, 99$ is $\frac{30+90}{2}$. The median of $10, 30, 50, 60, 90$ is 50.

10. (3 bonus points) Think of a *good*[5] exam or homework question for the material covered in Lecture 2

[3] Surprisingly no ship arrive without using a wormhole! Your galaxy is far far away from all the other galaxies.

[4] You can decide what data structure to use and what information you store in the data structure. You might choose to use more than on data structure to solve this problem.

[5] Only well thought out questions will receive the bonus points.