# CS6033 Homework Assignment 2*

Due Feb. 7th at 5:30 p.m.
Turn in this assignment as a PDF file on NYU classes
No late assignments accepted†

February 1, 2017

1. (15 points) For the following array

| 9 | 12 | 13 | 17 | 25 | 88 |
|---|---|---|---|---|---|

   (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 1a, 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 56 into the heap that resulted from question 1b. Show the array after you have inserted the item.

2. (10 points) You have a free weekend. Trying to fill up the time, you decide to watch as many movies as possible (bing movie watching!)

   Friends keep sending you suggestions. Some of these friends don't know you that well and may send a title you don't like. You want to optimize your weekend by watching the best movies suggested to you. You decide that Netflix knows you better than your friends (scary!).

   You want to watch the movies in the order that Netflix ranks them.

   If you are given $n$ movie titles, (and in $\Theta(1)$ time you can find Netflix's rank for you of any movie). Write an $\Theta(k \log(n) + n)$ algorithm so you watch the best $k$ movies out of $n$ movies. (Of course - you may use any algorithm we have discussed in class!)

3. (25 points) Consider of 4-ary max-heap. A 4-ary max-heap is like a binary max-heap, but instead of 2 children, nodes have 4 children.

   (a) How would you represent a 4-ary max-heap in a array?

   (b) What is the height of a 4-ary 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 $n$.

*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.

4. (20 points) Prove the correctness of `HEAP-DECREASE-KEY` using a loop invariant.

5. (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.

```
MAXIMUM(A, l, r)
1)  if (r − l == 0)
2)      return  A[r]
3)
4)  lmax = MAXIMUM(A, l, ⌊(l + r)/2⌋)
5)  rmax = MAXIMUM(A, ⌊(l + r)/2⌋ + 1, r)
6)  PRINT(rmax, lmax)
7)  if rmax < lmax
8)      return  lmax
9)  else
10)     return  rmax
```

   (c) (20 points) You host an online game, where there is a number of rounds. When a player finishes a round his or her score is added to a scoreboard. Those who have all their scores in the top 1/2 will be invited to play in the tournament. Thus it is important for people to quickly to know if they are in the top 1/2. This is easily done by comparison to the median. Design an algorithm to keep track of the median as items are added one at a time ($O(\log n)$ time for each inserted item) and $O(n)$ space. (Hint: use to binary heaps.)

   Reminder: The median of $1, 9, 21$ is 9. The median of $1, 3, 9, 21$ is $\frac{3+9}{2}$. The median of $1, 3, 9, 21, 56$ is 9.

   (d) (3 bonus points) Think of a *good*[2] exam question for the material covered in Lecture 2

[1] You do not need to solve the recurrence formula.

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