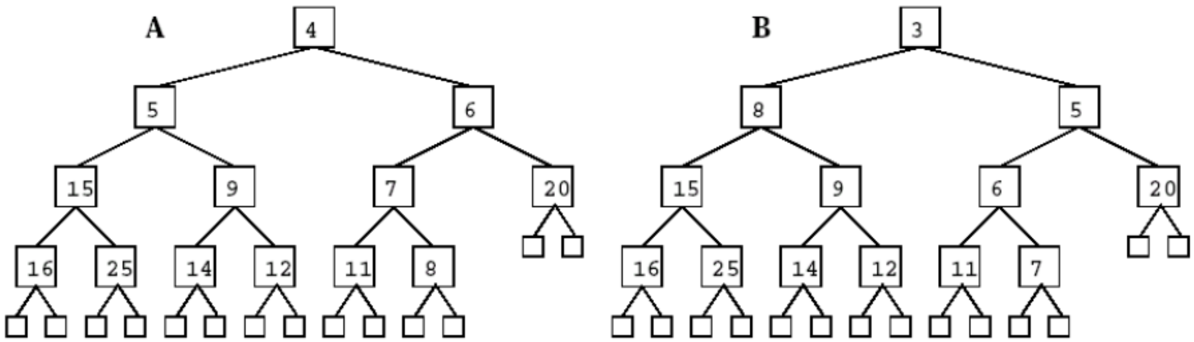
Algorithms and Programming I
Spring 2015
Exam#1 Review

Problem#1

(a) Give definition of a heap.
(b) What minimal sequences of ***insert*** and/or ***removeMin*** operations on heap ***A*** will transform it into heap ***B***? Draw the heap after each operation.



Problem#2

Solve the following recurrences using Master theorem:

(1) $T(n)= 6T(n/3) + \Theta(n^{\log_3 6})$
(2) $T(n)= 4T(n/2) + \Theta(n^2)$
(3) $T(n)= T(4n/5) + \Theta(n)$

Problem#3

What is the running time of these algorithms?

The Algorithm	Running time
Insertion sort	
Merge Sort	
Heap Sort	
Quick Sort	
Selection Sort	

Problem#4

How does the key in a node compare to the keys of its children in a max heap?

Problem#5

Rank the following functions by increasing order of growth; that is, find an arrangement g_1, g_2, g_3, g_4 of the functions satisfying $g_1 = O(g2)$, $g_2 = O(g_3)$, $g_3 = O(g_4)$.
(For example, the correct ordering of n^2, n^4, n, n^3 **is** n, n^2, n^3, n^4.)

$f1 = n^{\log n}$

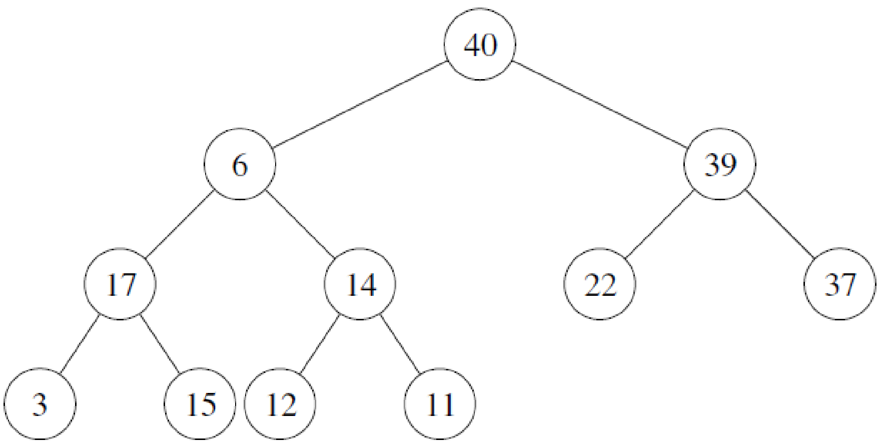
$f2 = \sqrt{n}$

$f3 = n^{3+\sin(n)}$

$f4 = \log n^n$

Problem#6

What is the max-heap resulting from performing on the node storing 6?



Problem#7

The following array is a max heap: [10,3, 5, 1, 4, 2].

Problem#8

In max-heaps, the operations insert, max-heapify, find-max, and findmin all take O(log n) time. (T,F)

Problem#9

In the merge-sort execution tree, roughly the same amount of work is done at each level of the tree. (T,F)

Problem#10

In a min-heap, the next largest element of any element can be found in O(log n) time. (T,F)

Problem#11

Solve the following recurrences using both recursion tree

$$T(n) = 3T(n/4) + n^2$$

Problem#12

Which of the two algorithms (Heap Sort, Merge Sort), implemented as described in class, is a better choice if space (memory usage) is the primary concern, rather than running time?

Problem#13

Suppose a binary max-heap contains 80 distinct keys. How many distinct positions might contain the smallest element in H?

Problem#14

Student X implemented Merge Sort, but due to a coding error, his implementation divided the input array of size n into a first 'half' of length n/ 3 and a second 'half' of length 2n/ 3 , which were (recursively) sorted and the results merged. The resulting output is correct. What is the asymptotic running time of Student X's algorithm?