Algorithms
Fall 2014

Problem#1

Describe an O (n log k)-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists.

Problem#2

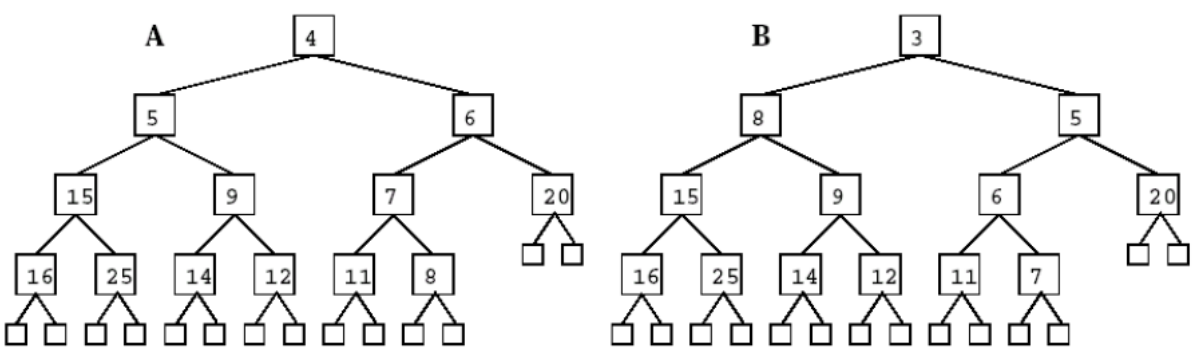
Define Big Omega, Big theta and Big Oh bounds of a function.

Problem#3

You are implementing an algorithm that draws part of the landscape of a terrain, and you are faced with the following problem: You are given the heights of N points of the terrain's grid, and you need to find and sort, as fast as possible, the $\sqrt{N}$ highest of them. Give an algorithm that does this, and argue that no one can do better (up to a constant factor, of course!) (Hint: Your algorithm should run in O (N) time; there is a simple argument why this is the best possible.)

Problem#4

(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#5

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#6

What is the running time of these algorithms?

The Algorithm	Running time
Insertion sort	
Merge Sort	
Heap Sort	

Problem#7

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

Problem#8

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, n2, n3, n4.)

$f1 = n^{\log n}$

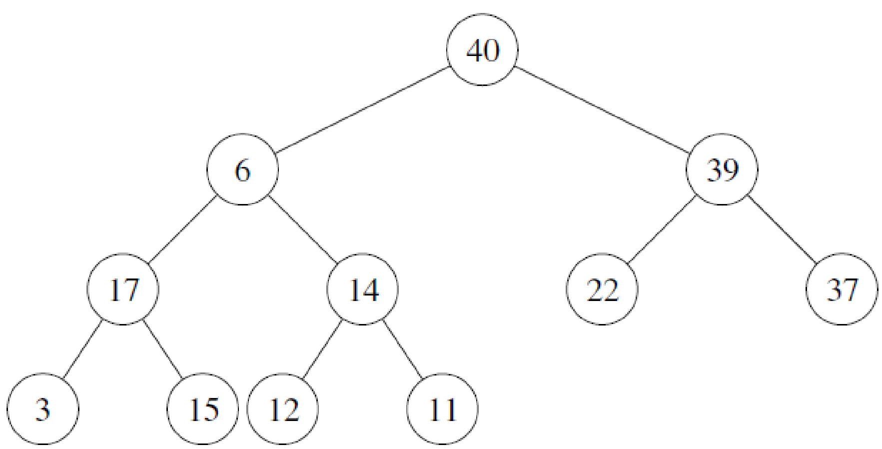
$f2 = \sqrt{n}$

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

$f4 = \log n^n$

Problem#9

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



Problem#10

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

Problem#11
In max-heaps, the operations insert, max-heapify, find-max, and findmin all take O(log n) time. (T,F)

Problem#12
In the merge-sort execution tree, roughly the same amount of work is done at each level of the tree. (T,F)

Problem#13
In a min-heap, the next largest element of any element can be found in O(log n) time. (T,F)

Problem#14
Solve the following recurrences using recursion tree:
$T(n) = 3T(n/4) + n^2$