HEAPS, SORTING, RECURRENCES & HASHING

(1) Draw the binary min heap that results from inserting 11, 9, 12, 14, 3, 15, 7, 8, 1 in that order into an initially empty binary heap.

(2) Solve the following recurrences: (1) $T(n) = 2T(n/8) + \sqrt[3]{n}$, (2) $T(n) = 3T(n/4) + n^2$.

(3) An array $A[1..n]$ is called wiggly if $A[1] \leq A[2] \geq A[3] \leq A[4] \geq A[5]$ and so on upto $A[n]$. Convert an unsorted array $B[1..n]$ into a wiggly array in $\Theta(n)$ time.

(4) Develop an algorithm that computes the kth smallest element of a set of n distinct integers in $O(n + klogn)$ time.

(5) Give the pseudo-code description for performing a removal from a hash table that uses linear probing to resolve collisions, where we do not use a special marker to represent deleted elements. That is, we must rearrange the contents of the hash table so that it appears that the removed entry was never inserted in the first place.

(6) Given an array A of n integers in the range $[0..n^2 - 1]$. Describe a simple method for sorting A in $O(n)$ time.

(7) [Challenge Problem] Given two sorted arrays A and B, each of size n. Find the kth smallest element in the union of elements in A and B in $O(logn)$ time.

(8) [Challenge Problem] Let there be n items labeled 1 to n and n positions labeled 1 to n. Let $T(n)$ be the number of ways in which these items can be arranged so that no item is at its own location (i.e., for all i, item numbered i can't be at location i). Find a recurrence for $T(n)$. Compute $T(6)$. Can you solve the recurrence for general n?