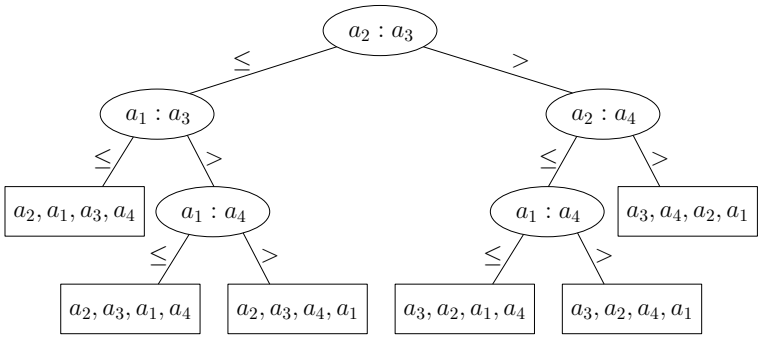
COMP 3711 Design and Analysis of Algorithms
2015 Fall
Solutions to Assignment 2

1. (a) $O(n \log n)$
   (b) Build the heap: $O(n)$. Call Extract-Min k times: $O(k \log n)$. Total time: $O(n + k \log n)$.
   (c) Build the heap with repeated insertions: $O(n \log n)$. Call Extract-Min k times: $O(k \log n)$. Total time: $O(n \log n)$.
   (d) We first use the linear-time selection algorithm to find the k-th smallest number (denoted as a_k), which takes $O(n)$ time. Then find all the k numbers smaller than or equal to a_k, which takes $O(n)$ time. Finally, we sort these k numbers, which takes $O(k \log k)$ time. So the total running time is $O(n + k \log k)$.

2. The subtree T_3:



3. We use counting sort to sort all strings by the first character. This takes $O(m)$ time. Note that when moving the strings around, we move only the pointers to the strings. This will divide the strings into groups such that within each group, all strings share the same first character. This completes the first iteration of the algorithm.

   Next, within each group, we first find all strings with length 1, and put them at the front of the group. Then we sort the remaining strings in the group by the second character by counting sort. These strings will thus be further divided into smaller groups such that within each group, all strings share the same first 2 characters. We do so for each group. This completes the second iteration of the algorithm. Then, we do the 3rd iteration on all these small groups. In the 3rd iteration for each group, we first find all strings with length 2, and put them at the front of the group, and then use counting sort to sort the remaining strings by the 3rd character. We repeat the iterations until the d-th iteration, where d is the maximum length of any string.

   Analysis: Each iteration takes time O(the number of strings that participate in this iteration). In the first iteration, all strings participate. In the 2nd iteration, those strings with 1 character will not participate; in the 3rd iteration, those with 2 characters will not. In general, a string with k characters participate in the first k iterations. So, the total running time is $\sum_s (\text{length}(s)) = O(n + m) = O(n)$.

4. Use `https://www.cs.usfca.edu/~galles/visualization/AVLtree.html` to check your answer.

5. This problem is the same as the birthday paradox, by viewing the n elements as people and the m table locations as days. Thus, the expected collision number is $\binom{n}{2}\frac{1}{m}$.