Please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.

2. One instruction per line

3. Match the brackets with a horizontal line

4. Number your lines

5. Write down if your array is indexed 0...n−1 or 1...n

**Problem 1** A $d$-ary heap is like a binary heap, but (with one possible exception) nonleaf nodes have $d$ children instead of 2 children.

1. How would you represent a $d$-ary heap in an array?

2. What is the height of a $d$-ary heap of $n$ elements in terms of $n$ and $d$? Explain your reasoning. A correct \( \Theta() \) estimate with explanation is worth 80% of the score here.

3. Give an efficient implementation (pseudocode is required) of EXTRACT-MAX in a $d$-ary max-heap. Analyze its running time in terms of $d$ and $n$.

4. Give an efficient implementation (pseudocode is required) of INSERT in a $d$-ary max-heap. Analyze its running time in terms of $d$ and $n$.

**Problem 2** Give an example of a binary search tree $T$ and two nodes $x$ and $y$ of $T$ such that deleting $x$ and then $y$ from $T$ leaves a different tree than the tree obtained by deleting $y$ and then $x$ from $T$. Draw $T$ and the tree after each deletion (use for deletion the algorithm from the notes).

**Problem 3** Consider the following sequence of keys:

\[(22,45,2,10,18,16,5,30,50,12,1)\]

Consider the insertion of items with this set of keys, in the order given, into an initially empty (2,3) tree $T$. Draw $T$ after each insertion.

**Problem 4** Suppose we are given two sorted arrays $S$ and $T$, each with $n$ items. Describe an $O(\log n)$-time algorithm for finding the $k$th smallest key in the union of the keys from $S$ and $T$ (assuming no duplicates). Present pseudocode, argue correctness and analyze the running time.