Consider the following functions for problems 1 and 2.

```
void selectionSort(int array[])
{
    sort(array, 0);
}

void sort(int[] array, int i)
{
    if (i < array.length - 1)
    {
        int j = smallest(array, i);
        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
        sort(array, i + 1);
     }
 }

 int smallest(int[] array, int j)
 {
     if (j == array.length - 1)
         return array.length - 1;
     int k = smallest(array, j + 1);
     return array[j] < array[k] ? j : k;
 }
```

1. Draw the recursion tree for `selectionSort` when it is called for an array of length 4. Show the activations of `selectionSort`, `sort` and `smallest` in the tree.

2. Determine a formula that counts the numbers of nodes in the recursion tree. What is Big-Θ for execution time? Determine a formula that expresses the height of the tree. What is the Big-Θ for memory?

   For problems 3 and 4, consider the following functions that implement the dequeue operation for a priority queue that is implemented with a heap.

```
int[] pQueue;
int length;

int dequeue()
{
    int node = 1;
    int value = pQueue[--length];
    int maxValue = pQueue[node];

    int location = sift(node * 2, value);
    pQueue[location] = value;
    return maxValue;
}

int sift(int node, int value)
{
```

```
    if (node <= length)
    {
        if (node < length && pQueue[node] < pQueue[node + 1])
            node++;
        if (value < pQueue[node])
        {
            pQueue[node / 2] = pQueue[node];
            return sift(node * 2, value);
        }
    }
    return node / 2;
}
```

3. Write the recurrence equation and initial conditions that expresses the execution time cost for the `sift` function in the above algorithm. Draw the recursion tree assuming that $n = 8$. Assume that n represents the number of nodes in the subtree at each step of the sifting operation.

4. Determine the critical exponent for the recurrence equation in problem 3. Apply the Little Master Theorem to solve that equation specifically stating which part of the theorem is applied to arrive at the solution.