Problem Set 3

Questions to understand recursive complexity analysis

Question 1

In the previous lesson, we implemented merge sort where we divided the array into two parts at each recursion step. Imagine you are asked to implement merge sort by dividing the problem into three parts instead of two. You can assume for simplicity that the input size will always be a multiple of 3. Use a priority queue to merge sub-arrays in the combine step.

a- Provide implementation for the 3-way division merge sort.

b- Provide a generalized expression for the number of recursion levels.

c- Work out the time complexity for the 3-way merge sort.

d- Will implementing merge sort by dividing the problem into a greater number of subproblems improve execution time?

Question 2

We implemented unoptimized code for generating permutations of a string. The solution used extra space and also ran the main loop for the entire length of the array on each invocation. Given the following optimized solution, can you work out the time complexity?

```
class Demonstration {
    public static void main( String args[] ) {
        char[] array = new char[] { 'a', 'b', 'c', 'd' };
        permutate(array, 0);
    }

    private static void swap(char[] str, int i, int j) {
        char temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }

    static void permutate(char[] str, int index) {
        // base case
        if (index == str.length) {
            System.out.println(str);
            return;
        }

        // regular case
        for (int i = index; i < str.length; i++) {
            swap(str, index, i);
            permutate(str, index + 1);
            swap(str, index, i);
        }
    }
}
```

Question 3

We are asked to implement a *n-dimensional* integer list. The constructor will take in the parameter *n* and the class will expose two methods `get(int n)` and `put(int n)`. If n equals 3, then we'll first create a list of 3 element, then initialize each element of first list with another list of 3 elements and finally initialize each element of the second list with a list of 3 elements. The get and put functions act on the elements of the innermost list.

As an example, with n=3, we'll get a total of 27 elements. The get and put methods should be able to specify index from 0 to 26.

a- Can you code the class?

b- What is the time complexity of initializing the list?

c- What is the time complexity of the get and put methods?

d- What is the space complexity?