CS202 Assignment 1 – Shin Minchul

Q1. Study of Choose.java

This given Java program works similar to Combination in Mathematics, where n stands for total number of objects in the set, r stands for the number of chosen objects from the set. The program prints all possible subsets of size r from an array e of n elements. The method "choose (int b, int c)" generates recursion to recurse all possible selection of r elements from the given n elements that is inputted by the user.

```
-Assignment\Assignment1>java Choose 6 3
2 1 0
3 1 0
4 1 0
5 1 0
3 2 0
4 2 0
5 2 0
4 3 0
5 3 0
5 4 0
3 2 1
5 2 1
5 2 1
4 3 1
5 3 1
5 4 1
4 3 2
5 3 2
5 4 2
5 4 3
```

Figure 2. Example output of Choose.java

As the key implementation of the program is related to choosing method, I will further explain about it. The method looks like working as follows:

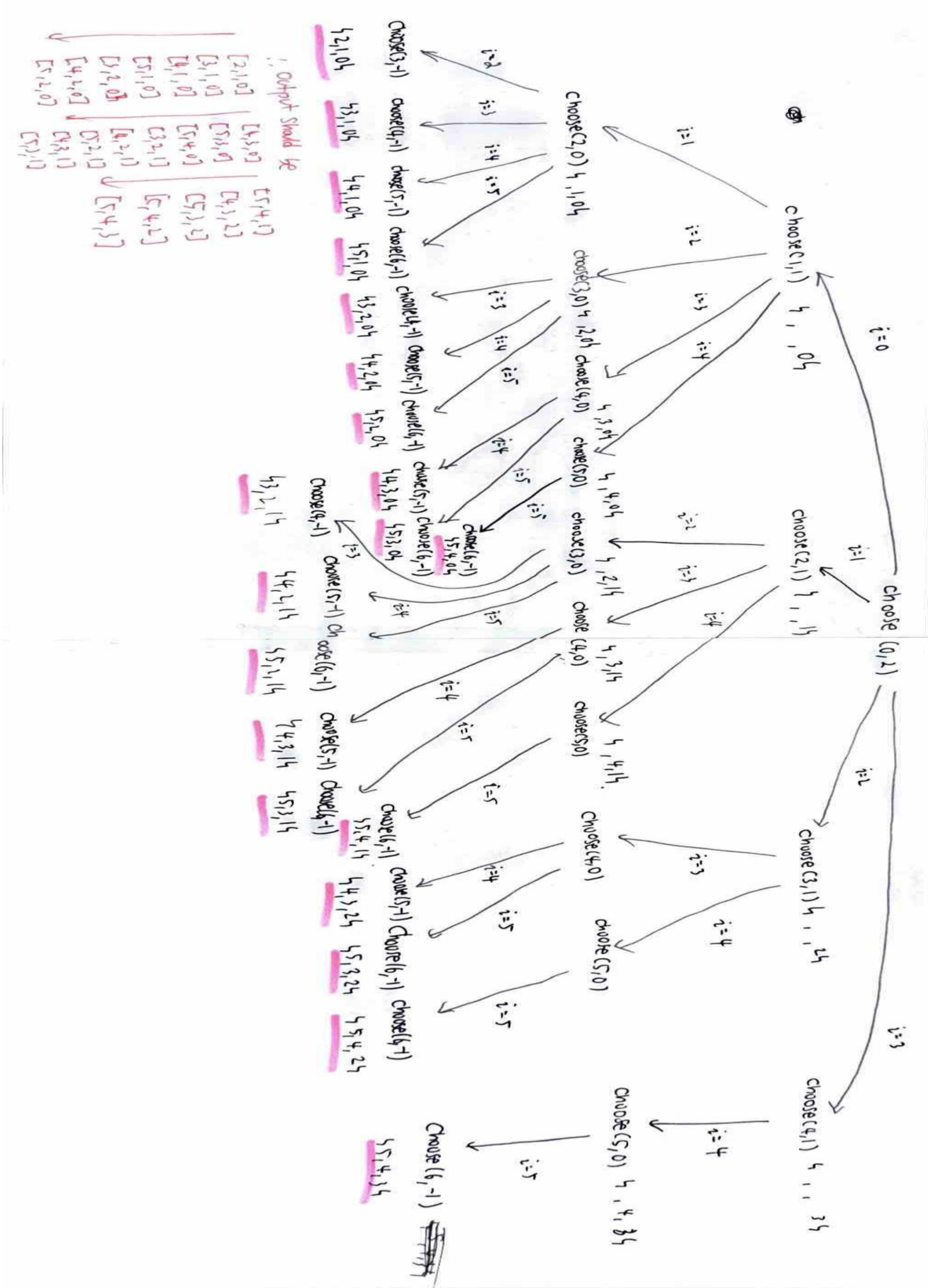
1. It selects r elements from n.
2. If c < 0, it prints the selected subset 'a' array.
3. Else, it iterates over possible selection from i = b to i < n – c.
4. Each selection in each recursive level will be stored in array a, and recursive call is made for the next element.

```java
    public static void process(int a[]) {
        // Print the generated subset
        for (int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }

    public static void choose(int b, int c) {
        // BASE CASE: If c < 0, print the selected subset stores in array
        if (c < 0)
            process(a);
        else
            // Iterate through the elements starting from index 'b'
            for (int i = b; i < n - c; i++) {
                a[c] = e[i]; // Select element and store it in a[c]
                choose(i + 1, c - 1); // Recursive call for next element
            }
    }
```

Figure 3. Comments and explanation of what the function does

***Just in case it is blurred, the image is saved in pdf file inside the folder**



Q3. Asymptotic Analysis

3.1 Part 1

Using the Master Theorem, we now calculate Asymptotic bound for the recurrence:

$$T(n) = 12 \cdot T\left(\frac{n}{4}\right) + n^2$$

First, we should get a, b and f(n) to apply for the Master Theorem which are the following:

$$a = 12, b = 4, and\ f(n) = n^2$$

From here, calculate $\log_b a = \frac{\log(12)}{\log(4)} \approx 1.792$ and compare $n^{\log_b(a)} = n^{1.792}$ with $f(n) = n^2$.

As $n^2 > n^{1.792}$, f(n) grows faster. Thus, Case 3 may be applied. Let's verify the Regularity Condition, it must hold:

$$12\ f\left(\frac{n}{4}\right) \le c \cdot f(n) \text{ for some c } < 1, \text{and sufficiently large n}$$

$$12\ f\left(\frac{n}{4}\right) = 12\left(\frac{n}{4}\right)^2 = 12 \cdot \frac{\mathrm{n}^2}{16} = \frac{12}{16}\mathrm{n}^2 = \frac{3}{4}\mathrm{n}^2 \text{ (LHS)}$$

$$\frac{3}{4}n^2 \le cn^2$$

The equation satisfied for some c < 1 such as 0.5 hence, case 3 applied to conclude: $T(n) = \Theta(n^2)$

3.2 Part 2

Using the Master Theorem, we now calculate Asymptotic bound for the recurrence:

$$T(n) = 3 \cdot T\left(\frac{n}{3}\right) + n \log n$$

$$a = 3, b = 3, and\ f(n) = n \log n$$

From here, calculate $\log_b a = \frac{\log(3)}{\log(3)} = 1$ and compare $n^{\log_b(a)} = n^1$ with $f(n) = n \log n$. Even though the n log n grows slightly faster than n^1, to apply case 3 it must hold $\mathrm{f(n)} = \Omega(\mathrm{n}^{1+\varepsilon})$. So, following equation should hold:

$$n\ log\ n = \Omega(n^{1+\varepsilon})$$

However, for any small $\varepsilon > 0$, we see $n\ log\ n = o(n^{1.1})$. Hence, $n\ log\ n$ grows slower than $\mathrm{n}^{1+\varepsilon}$, thus instead of case 3, check whether it is form of case 2 condition: $\Theta(n^{log_b a} \cdot \log^k n)$. Here it satisfies the condition 2, and f(n) = n log n and k = 1 hence,

$$T(n) = \Theta(n \log^2 n)$$

- END OF THE ANSWER -