Homework 3: Recursion

I. *Lập và giải công thức đệ quy xác định độ phức tạp thuật toán*

1.

2.4.1.a.

$X(1) = 1$

$X(n) = 2^2 x(n-6)$

$= 2^k x(n-3k)$

$O(2^n)$

2.4.1.b.

Ta có: $x(n) - x(n-2) = 2$

$x(n-2) + x(n-4) = 2$

$x(n-4) - x(n-6) = 2$

…

$x(3) - x(1) = 2$

Cộng 2 vế của các đẳng thức trên, ta được:

$x(n) - x(n-2) + x(n-2) + x(n-4)x(n-4) - x(n-6) + ... + x(3) - x(1)$

$= 2 * n/2$

$\Leftrightarrow x(n) - x(1) = n$

$\Leftrightarrow x(n) = n + x(1)$

$\Leftrightarrow x(n) = n$

2.4.10.

Thuật toán trên được dùng để kiểm tra xem 1 đồ thị được định nghĩa bởi ma trận kề có phải là hoàn chỉnh không.

Worst case: thuật toán cần kiểm tra hết tất cả các cạnh của đồ thị để xác định. Nó sử dụng đệ quy để kiểm tra tính hoàn chỉnh của nó vì vậy mỗi cạnh nó chỉ xét đúng 1 lần. Tổng số cạnh mà nó phải kiểm tra là: $\frac{n*(n-1)}{2}$ (n đỉnh)

Cơ sở n= 1. Với n>1 nó sẽ đệ quy với ma trận kích thước (n-1)x(n-1) sau đó xét n – 1 lần. Do đó độ phức tạp sẽ là $O(n^2)$

II. *Lập trình đệ quy*

Foder Coding/Bai2

III. *Đặt bài toán, thiết kế, phân tích và triển khai thuật toán*

Đề bài toán: Tính tổng các phần tử trong một mảng số nguyên

Phân tích bài toán: Để thực hiện bài toán bằng đệ quy ta có thể tính tổng các phần tử con của mảng

Xây dựng thuật toán:

+ Nếu mảng chỉ có 1 phần tử thì trả về giá trị của phần tử đó.

+ Nếu mảng có nhiều hơn 1 phần tử, tính tổng của phần tử đầu tiên và sau đó từ phần tử thứ 2 trở đi tính tổng bằng cách gọi đệ quy hàm tính tổng.

Chứng minh tính đúng: Quy nạp: Giả sử thuật toán đúng với mảng kích thước n – 1 ta cần chứng minh thuật toán đúng với mảng kích thước n.

Nếu n = 1, sumArr = giá trị của phần tử đó (luôn đúng)

Nếu n > 1, Ta tính tổng của phần tử đầu tiên và phần tử trong mảng con

sumSubArr = arr[1] + arr[n-1] (Theo điều giả sử luôn đúng)

⇨ sumArr = sumSubArr + arr[1]

⇨ Ta có đpcm

Độ phức tạp của thuật toán: O(n). Vì thuật toán sẽ thực hiện n – 1 phép cộng, vì mỗi lần đệ quy thì phần tử đầu tiên trong mảng sẽ được cộng vào tổng do đó số phép cộng thực cần thực hiện sẽ giảm đi 1 đơn vị

Coding: Foder Coding/Bai3/SumArrRecurive.java

```java
public class SumArrRecursive {
    2 usages
    public static int sumRecursive(int[] arr, int n) {
        if (n == 1)
            return arr[0];
        else
            return arr[n - 1] + sumRecursive(arr, n: n - 1);
    }

}
```