

Problem Set 1: Big-Oh & Struktur Data Dasar

[Big-Oh]

1. Tentukan performa Big-oh dari potongan kode program berikut:

```
int c = 0;
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++){
        c = c + 1;
    }
}
```

2. Tentukan performa Big-oh dari potongan kode program berikut:

```
int c = 0;
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++){
        count = count + 1;
    }
}
```

3. Tentukan performa Big-oh dari potongan kode program berikut:

```
int i = n;
int c = 0;
while (i > 0){
    c = c + 1;
    i = i / 2; //Note: integer division
}
```

4. Tentukan performa Big-oh dari potongan kode program berikut:

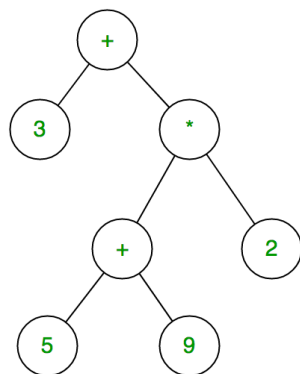
```
int c = 0;
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++){
        for (int k = 0; k < n; k++){
            c = c + 1;
        }
    }
}
```

5. Tentukan performa Big-oh dari potongan kode program berikut:

```
int c = 0;
for (int i = 0; i < n; i++){
    c = c + 1;
}
for (int j = 0; j < n; j++){
    c = c + 1;
}
for (int k = 0; k < n; k++){
    c = c + 1;
}
```

[Struktur Data]

1. Buat program menggunakan struktur Stack untuk melakukan pengecekan terhadap keseimbangan symbol berikut
 - a. Bahasa Pascal (begin/end, (), [], {})
 - b. C++ (/ * *, (), [], {})
 - c. Jika error, maka program perlu menginfokan alasannya
2. Buatlah program menggunakan struktur Stack jika inputnya adalah ekspresi aritmatika dalam string seperti $(2+3)*3-(10/5)$ dan output adalah hasil perhitungan dari ekspresi tersebut!
3. Buatlah program menggunakan struktur Tree jika inputnya adalah ekspresi aritmatika dalam string seperti $3+((5+9)*2)$ menjadi Tree seperti pada gambar di kiri. Algoritma untuk



membangun Tree tersebut adalah sebagai berikut:

- a. Ubah ekspresi aritmatika menjadi notasi postfix (contoh: $3+((5+9)*2) \rightarrow 359+2*+$).
- b. Baca ekspresi aritmatika dalam bentuk postfix lalu push ke Stack jika yang dibaca adalah operan
- c. Jika menemukan operator maka pop 2 buah operan dan buat kedua operan tersebut sebagai child dari operator tersebut.