

Nama : Fajri Fahreza Azeta

Nim : 11850110334

Kls : TIF / 4B

Strategi Algoritma UAS

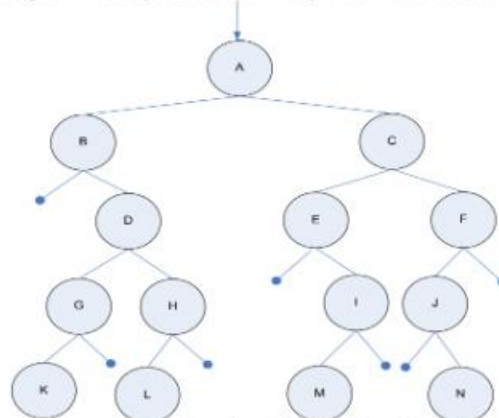
UJIAN AKHIR SEMESTER GENAP 2019/2020
JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI UIN SUSKA RIAU
Mata Kuliah : Strategi Algoritma
Semester/ Kelas : IV (A, B, C)

1. Perhatikan algoritma *brute force* berikut ini :

```
public static int secretAlgorithm(int[]A)
{
    int maxSum=0;
    for(int i=0;i<A.length;i++)
    {
        for(int j=i;j<A.length;j++)
        {
            int thisSum=0;
            for(int k=i;k<=j;k++)
            {
                thisSum +=A[k];
                if (thisSum>maxSum)
                    maxSum=thisSum;
            }
        }
    }
    return maxSum;
}
```

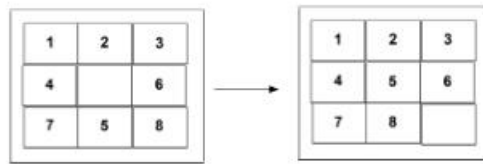
Iterasi (k) dilakukan sebanyak n dalam iterasi (j) sebanyak n dalam iterasi (i) sebanyak n. Sehingga kompleksitasnya $O(n^3)$. Desainlah algoritma diatas sehingga menghasilkan kompleksitas kuadratik ($O(n^2)$) ! (Bobot 40%)

2. (Divide and Conquer – Binary Search & Binary Tree Traversal, bobot 20%)



Tentukan urutan kunjungan terhadap simpul pada Binary Tree berikut berdasarkan pre-order, in-order, dan post-order!

3. (Decrease and Conquer – DFS & BFS, bobot 40%)
Perhatikan 8-puzzle berikut ini!



Jika open list dan close list DFS direpresentasikan dalam stack, dan BFS dengan struktur data queue, bangunlah pohon penelusurannya!

JAWABAN:

NAMA: Fajri Fahreza Azeta
KLS: TIF 4B
Nim: 11850110334
UAS - SA

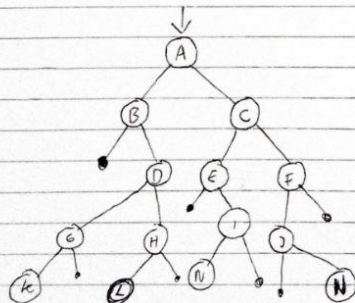
- ① Desain algoritma menghasilkan kompleksitas kuadratik:

```

public static int Secret Algorithm (int[] A)
{
    int maxSum = 0;
    for (int i = 0; i < A.length; i++)
    {
        int thisSum = 0;
        for (int j = i; j < A.length; j++)
        {
            thisSum += A[j];
            if (thisSum > maxSum)
            {
                maxSum = thisSum;
            }
        }
    }
    return maxSum;
}

```

②



Tentukan urutan kunjungan terhadap simpul pada Binary Tree tersebut

Pre-order = A, B, D, G, K, H, L, C, E, I, M, F, J, N

In-order = B, K, G, D, L, H, A, E, M, I, C, J, N, F

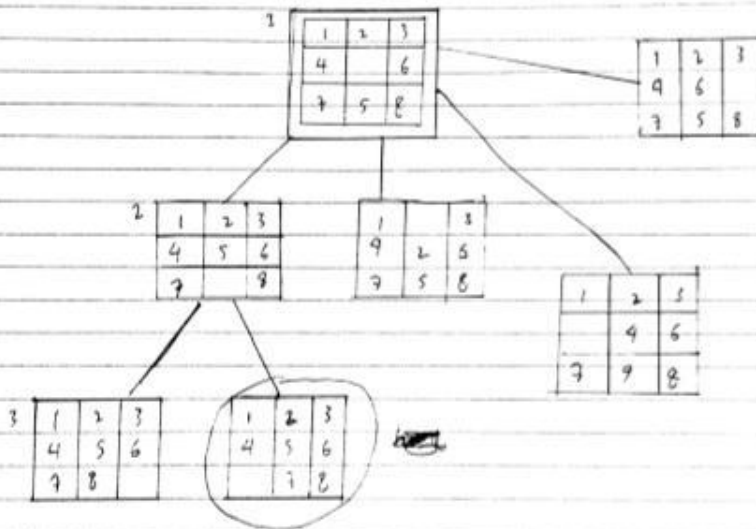
Post-order = K, G, L, H, D, B, M, I, E, N, J, F, C, A

3

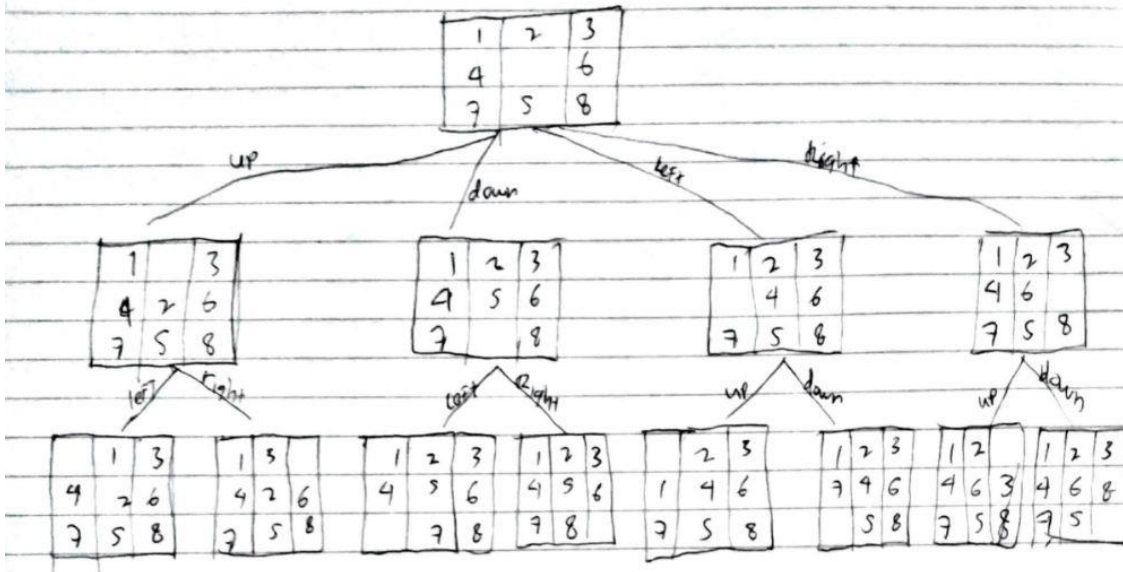


Dikenal Open List dan Close List, DFS direpresentasikan dalam Search dan BFS dengan struktur queue, bagaimana pohon penelusurannya

DFS (stack)



BFS - Queue



Mark list

