Praktikum Blind Search
(BFS dan DFS)

LATIHAN SOAL

A. 1. Jelaskan algoritma BFS !
2. Jelaskan algoritma DFS !

B. Aplikasi Game "Petani – Angsa – Serigala - Padi"

1. Tentukan ruang permasalahan (problem space) dari Game "Petani – Angsa – Serigala - Padi".
    a. Definisikan state (ruang keadaan) dari game tersebut.
    b. Tentukan state awal
    c. Tentukan state akhir
    d. Operator yang digunakan.
    e. Sebutkan aturan-aturan yang digunakan
2. Buatlah Graph untuk menggambarkan ruang keadaan (State Space) dari Game tersebut.
3. Carilah solusi dari Game tersebut menggunakan algoritma BFS dan DFS !

C. Aplikasi Game "riverIQGame"

1. Tentukan ruang permasalahan (problem space) dari Game **"Petani – Angsa – Serigala - Padi".**
    a. Definisikan state (ruang keadaan) dari game tersebut.
    b. Tentukan state awal
    c. Tentukan state akhir
    d. Operator yang digunakan.
    e. Sebutkan aturan-aturan yang digunakan
2. Buatlah Graph untuk menggambarkan ruang keadaan (State Space) dari Game tersebut.
3. Carilah solusi dari Game tersebut menggunakan algoritma BFS dan DFS !

D. Game Eight – Puzzle

1. Tentukan ruang permasalahan (problem space) dari Game **"Petani – Angsa – Serigala - Padi".**
    a. Definisikan state (ruang keadaan) dari game tersebut.
    b. Tentukan state awal
    c. Tentukan state akhir
    d. Operator yang digunakan.
    e. Sebutkan aturan-aturan yang digunakan

2. Buatlah Graph untuk menggambarkan ruang keadaan (State Space) dari Game tersebut sampai kedalaman 3 dengan node awal "143 706 582".

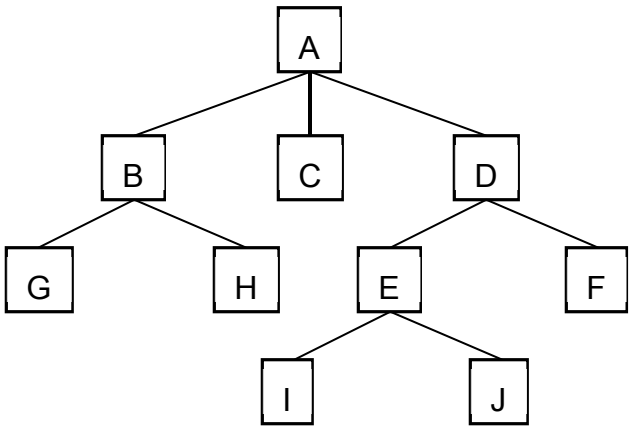
3. Dari graph tersebut carilah penyelesaian menggunakan algoritma BFS (Tentukan node tujuan pada kedalaman 3) !
4. Dari graph tersebut carilah penyelesaian menggunakan algoritma DFS (Tentukan node tujuan pada kedalaman 3, cari node pertama di kedalaman 3) !

PRAKTIKUM

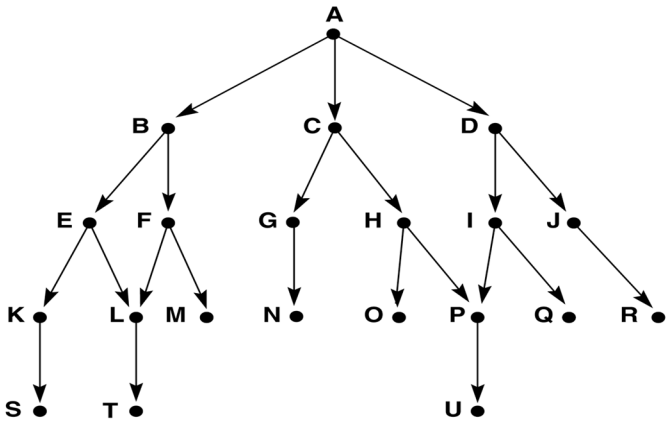
A. Algoritma BFS dan DFS pada Graph.

Diberikan contoh program BFS dan DFS.

1. Pada program terdapat class Vertex dan Class Graph. Jelaskan kegunaan dari:
   - Class Vertex
   - Class Vertex variable label, wasVisited
   - Class Graph
   - Class Graph variable vertexList[], adjMat[][], openQueue, openStack.
   - Class Graph method : addVertex(), addEdge(),getAdjUnvisitedVertex().
2. Terdapat tree seperti di bawah ini. Bagaimana program di fungsi `main()` ! Bagaimana output pencarian BFS dan DFS !



3. Contoh Program DFS apakah dapat menyelesaikan tree di bawah ini, dimana terdapat node yang sama, sehingga pada algoritma DFS/BFS terdapat langkah-langkah :
   Discard children of X if already on open or closed.
   Tambahkan pada program fungsi tersebut.



1. **open = [A]; closed = [ ]**
2. **open = [B,C,D]; closed = [A]**
3. **open = [C,D,E,F]; closed = [B,A]**
4. **open = [D,E,F,G,H]; closed = [C,B,A]**
5. **open = [E,F,G,H,I,J]; closed = [D,C,B,A]**
6. **open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]**
7. **open = [G,H,I,J,K,L,M]** (as L is already on open); **closed = [F,E,D,C,B,A]**
8. **open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]**
9. and so on until either U is found or **open** = [ ]

B. Game 8 Puzzle

Terdapat contoh program penyelesaian Game 8 Puzzle menggunakan BFS.

1. Modifikasilah program EightPuzzleBFS.java sehingga dapat berjalan seperti di bawah ini!
   Contoh 1
   Menyelesaikan permainan puzzle dengan state awal 120453786 dan state tujuan 123456780. Penyelesaian EightPuzzle dengan algoritma BFS , Tree yang terbangun sampai level 2 dan node yang dilalui sebanyak 4.

   Input :

```
EightPuzzleBFS eight = new EightPuzzleBFS("120453786", "123456780");
```

   Output :

```
1 0 120453786
2 1 123450786 down
3 1 102453786 left
4 2 123456780 down
Solution Exists at Level 2 of the tree

**********SOLUSI**********
****************************************
asal : 120453786
```

```
tujuan : 123456780

2 123456780 down
1 123450786 down
0 120453786
```

Contoh 2 :
Menyelesaikan permainan puzzle dengan state awal 123458670 dan state tujuan 123456780. Penyelesaian EightPuzzle dengan algoritma BFS , Tree yang terbangun sampai level 12 dan node yang dilalui sebanyak 1823.

Input

```
EightPuzzleBFS eight = new EightPuzzleBFS("123458670","123456780");
```

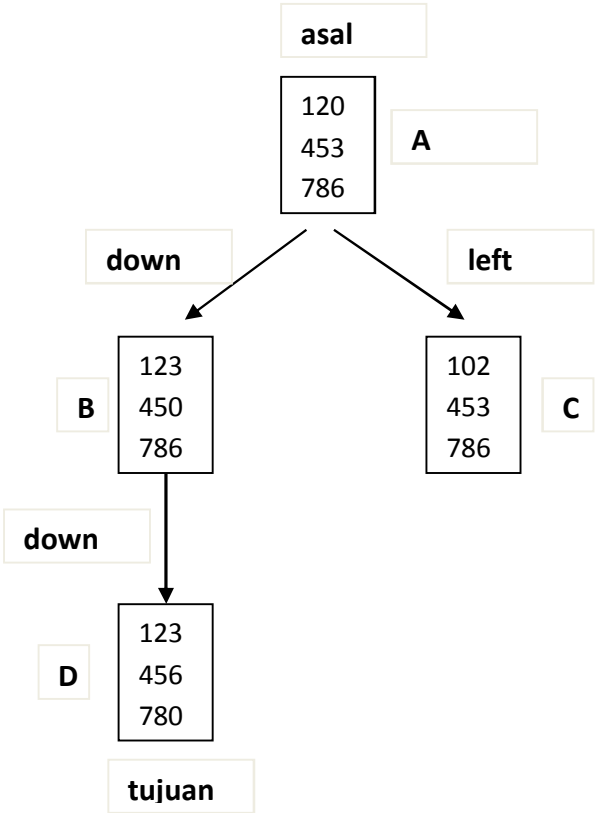
Output :

```
1 0 123458670
2 1 123450678 up
3 1 123458607 left
4 2 120453678 up
5 2 123405678 left
6 2 123408657 up
7 2 123458067 left
8 3 102453678 left
9 3 103425678 up
10 3 123475608 down
11 3 123045678 left
12 3 103428657 up
13 3 123048657 left
14 3 123480657 right
15 3 123058467 up
16 4 152403678 down
17 4 012453678 left
18 4 013425678 left
19 4 130425678 right
20 4 123475068 left
….
1816 12 512603478 right
1817 12 513426078 down
1818 12 136528470 down
1819 12 136502478 left
1820 12 023176548 up
1821 12 123706548 right
1822 12 123406758 up
1823 12 123456780 right
```

```
Solution Exists at Level 12 of the tree

************SOLUSI**********************
****************************************
asal : 123458670
tujuan : 123456780

12 123456780 right
11 123456708 right
10 123456078 down
9 123056478 left
8 123506478 left
7 123560478 up
6 123568470 right
5 123568407 down
4 123508467 right
3 123058467 up
2 123458067 left
1 123458607 left
0 123458670
```



Bantuan pemrograman.

Buatlah class Node yang merepresentasikan sebuah Node pada tree dengan informasi :

- String strParent : state puzzle dari parent node tersebut.
- String strChild : state puzzle dari node tersebut
- String operator : up, down, left dan right
- int level : kedalaman tree.

Node A (Node asal)	Node B

strParent="";
strChild="120453786" / A;
operator="";
level=0;
	strParent="120453786" / A;
	strChild="123450786" / B;
	operator="down";
	level=1;

Node C	**Node D (Node tujuan)**

strParent="120453786" / A;
strChild="102453786" / C;
operator="left";
level=1;
	strParent="123450786" / B;
	strChild="123456780" / D;
	operator="down";
	level=2;

2. Buatlah **program menggunakan java** untuk penyelesaian Puzzle menggunakan metode **Depth First Search (DFS).**
3. Buatlah Analisa Algoritma Pencarian Blind menggunakan algoritma BFS dan DFS dengan studi kasus EightPuzzle.

No	State awal EightPuzzle	State akhir EightPuzzle	Solusi ditemukan di level	Node yang di lalui	Solusi dari awal – akhir.
1					
2					
3					
4					
5					

Analisa Algoritma Pencarian Blind menggunakan algoritma BFS dan DFS :

(beri penjelasan)