

PRAKTIKUM KECERDASAN BUATAN



VRAYOGA LOREANSA/3122522005

ALBI NUR ROSIF/3122522010

MUHAMMAD NABIL ROYYAN/3122522027

TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

PENS PSDKU SUMENEP

D3 TEKNIK INFORMATIKA

1.

1. Selesaikan menggunakan algoritma Breadth First Search (BFS)

Kerjakan soal dibawah ini menggunakan algoritma BFS. Kerjakan dengan menggambar, langkah dimulai dengan state awal sampai dengan state akhir, perpindahan state dengan operator UP, DOWN, LEFT, RIGHT. Lakukan langkah ini sampai jumlah state 10.

Algoritma Pencarian Blind	Initial State	state akhir	Output																		
NRP Ganjil Breadth First Search (BFS)	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>8</td><td></td><td>6</td></tr><tr><td>7</td><td>4</td><td>5</td></tr></table>	1	2	3	8		6	7	4	5	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td></td></tr></table>	1	2	3	4	5	6	7	8		Open = [] Close = []
1	2	3																			
8		6																			
7	4	5																			
1	2	3																			
4	5	6																			
7	8																				

Jawaban :

1. Selesaikan menggunakan algoritma Breadth First Search (BFS)

Kerjakan soal dibawah ini menggunakan algoritma BFS. Kerjakan dengan menggambar, langkah dimulai dengan state awal sampai dengan state akhir, perpindahan state dengan operator UP, DOWN, LEFT, RIGHT. Lakukan langkah ini sampai jumlah state 10.

Algoritma Pencarian Blind	Initial State	state akhir	Output																		
NRP Ganjil Breadth First Search (BFS)	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>8</td><td></td><td>6</td></tr><tr><td>7</td><td>4</td><td>5</td></tr></table>	1	2	3	8		6	7	4	5	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td></td></tr></table>	1	2	3	4	5	6	7	8		Open = [] Close = []
1	2	3																			
8		6																			
7	4	5																			
1	2	3																			
4	5	6																			
7	8																				

2.

2. Gunakan program BFS mengenai puzzle, jalankan program tersebut. Lakukan 5 percobaan dengan merubah state awal atau state akhir, lalu catatlah hasil percobaan tersebut

	state awal	state akhir	BFS menemukan solusi pada level	Jumlah node
1				
2				
3				
4				
5				

Jawaban :

```

src > J Test.java > Test > asal
1  import java.util.HashMap;
2  import java.util.LinkedList;
3  import java.util.List;
4  import java.util.Map;
5  import java.util.Queue;
6
7  public class Test {
8
9      String asal, tujuan;
10     Queue<List> openQueue = new LinkedList<List>();
11     Map<String, Integer> map = new HashMap<String, Integer>(); // HashMap is used to ignore repeated nodes
12
13     List temp = new LinkedList<String>();
14
15     public Test(String asal, String tujuan) {
16         this.asal = asal;
17         this.tujuan = tujuan;
18     }
19
20     void up(List node) {
21         String str = (String) node.get(index:0);
22         int a = str.indexOf(str:"0");
23         if (a > 2) {
24             String s = str.substring(beginIndex:0, a - 3) + "0" + str.substring(a - 2, a) + str.charAt(a - 3)
25                 + str.substring(a + 1);
26             // System.out.print("(" + (map.get(str) + 1) + ", " + s + ")");
27             // System.out.print(map.get(str) + 1);
28             // System.out.println(" up " + s);
29             Integer level = (Integer) node.get(index:2) + 1;
30
31             addQueue(s, operator:"up", level);
32         }
33         // else
34         // System.out.println("(" + (map.get(str) + 1) + ", " + s + ") up tidak ada");
35     }
36
37     void down(List node) {
38         String str = (String) node.get(index:0);
39         int a = str.indexOf(str:"0");
40         if (a < 6) {
41             String s = str.substring(beginIndex:0, a) + str.substring(a + 3, a + 4) + str.substring(a + 1, a + 3) + "0"

```

```

42         + str.substring(a + 4);
43         // System.out.print("(" + (map.get(str) + 1)+", "+b+")");
44         // System.out.print(map.get(str) + 1);
45         // System.out.println(" down " + s);
46
47         Integer level = (Integer) node.get(index:2) + 1;
48         addQueue(s, operator:"down", level);
49     }
50 }
51 // else
52 // System.out.println("(" + (map.get(str) + 1)+", "+b+") down tidak ada");
53 }
54
55 void left(List node) {
56     String str = (String) node.get(index:0);
57     int a = str.indexOf(str:"0");
58     if (a != 0 && a != 3 && a != 6) {
59         String s = str.substring(beginIndex:0, a - 1) + "0" + str.charAt(a - 1) + str.substring(a + 1);
60         // System.out.print("(" + (map.get(str) + 1)+", "+b+")");
61         // System.out.print(map.get(str) + 1);
62         // System.out.println(" left " + s);
63
64         Integer level = (Integer) node.get(index:2) + 1;
65         addQueue(s, operator:"left", level);
66     }
67     // else
68     // System.out.println("(" + (map.get(str) + 1)+", "+b+") left tidak ada");
69 }
70
71 void right(List node) {
72     String str = (String) node.get(index:0);
73     int a = str.indexOf(str:"0");
74     if (a != 2 && a != 5 && a != 8) {
75         String s = str.substring(beginIndex:0, a) + str.charAt(a + 1) + "0" + str.substring(a + 2);
76         // System.out.print("(" + (map.get(str) + 1)+", "+b+")");
77         // System.out.print(map.get(str) + 1);
78         // System.out.println(" right " + s);

```

```

78         // System.out.println("right " + 0);
79
80         Integer level = (Integer) node.get(index:2) + 1;
81         addQueue(s, operator:"right", level);
82     }
83
84     // else
85     // System.out.println("(" + (map.get(str) + 1)+","+b+") right tidak ada");
86 }
87
88 public void bfs() // breadth-first search
89 {
90
91     // untuk menambahkan node awal
92     addQueue(asal, operator:"", level:0);
93     // untuk mengecek apakah queue tidak empty
94     int no = 0;
95     while (openQueue.peek() != null) {
96
97         List X = openQueue.remove();
98         System.out.println(X.get(index:2) + " " + X.get(index:1) + " " + X.get(index:0));
99         if (X.get(index:0).equals(tujuan)) {
100             System.out.println("Solution Exists at Level " + X.get(index:2) + " of the tree");
101             System.out.println("jumlah " + no + " node");
102             break;
103         } else {
104
105             // generate children X
106             up(X); // Move the blank space up and add new state to queue
107             down(X); // Move the blank space down
108             left(X); // Move left
109             right(X); // Move right and remove the current node from Queue
110         }
111         no++;
112     }
113 }
114

```

```

115 // menambahkan string baru ke queue dan map
116 // untuk dilakukan pengecekan terlebih dahulu
117 // Add method to add the new string to the Map and Queue
118 void addQueue(String str, String operator, int level) {
119     if (!map.containsKey(str)) {
120         map.put(str, level);
121         // openQueue.add(str);
122         temp = new LinkedList<String>();
123         temp.add(str);
124         temp.add(operator);
125         temp.add(level);
126         openQueue.add(temp);
127     }
128 }
129
130 Run | Debug
131 public static void main(String[] args) {
132     // Test_DFSKu eight = new
133     // Test_DFSKu("120453786","123456780");
134     // Test_DFSKu eight = new
135     // Test_DFSKu("123458670","120453678","145678902"); //OK
136     Test eight = new Test(asal:"073845621", tujuan:"123456780"); // OK
137     eight.bfs();
138 }

```

Output :

No. 1

```
22 down 123456780  
Solution Exists at Level 22 of the tree  
jumlah 88691 node  
PS D:\file yoga\kuliah\semester 4\kecerdasan buatan\coba1>
```

No.2

```
Solution Exists at Level 26 of the tree  
jumlah 157732 node  
PS D:\file yoga\kuliah\semester 4\kecerdasan buatan\coba1>
```

No.3

```
Solution Exists at Level 24 of the tree  
jumlah 138721 node  
PS D:\file yoga\kuliah\semester 4\kecerdasan buatan\coba1>
```

No.4

```
Solution Exists at Level 23 of the tree  
jumlah 105574 node  
PS D:\file yoga\kuliah\semester 4\kecerdasan buatan\coba1>
```

No.5

```
Solution Exists at Level 25 of the tree  
jumlah 145069 node  
PS D:\file yoga\kuliah\semester 4\kecerdasan buatan\coba1>
```

Tabel :

	State awal	State akhir	BFS menemukan solusi pada level	Jumlah node
1	073845621	123456780	Level 22	88691
2	324516870	123456780	Level 26	157732
3	834651270	012345678	Level 24	138721
4	724680153	123456780	Level 23	105574
5	012345678	724680153	Level 25	145069