

Initial State 1:

Memory usage: 442.171875KB

Runtime: 4 Milliseconds

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with files `bfs.java`, `bfs2.java`, `bfs3.java`, and `iddfs.java`.
- Editor:** Displays the source code of `iddfs.java`. The code is a recursive depth-first search algorithm for finding a path in a 15-puzzle state space. It starts with an initial state `["1","2","3","4","5","6","7","8","9","10","11","13","14","15","0"]` and a goal state `["1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","0"]`.
- Console:** Shows the output of the program. It prints a sequence of numbers representing the path found: `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 11, 13, 14, 0, 12]`, `[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 10, 11, 13, 14, 15, 12]`, `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 13, 14, 15, 12]`, `[1, 2, 3, 4, 5, 6, 7, 0, 9, 10, 11, 8, 13, 14, 15, 12]`, and `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0]`. It then prints `Solution found.`, `Memory usage: 442.171875KB`, and `Runtime: 4 Milliseconds`.

Initial State 2:

The screenshot shows the Java VisualVM interface monitoring the `iddfs` application (pid 2380). The interface includes several monitoring panels:

- Overview:** Shows the application's uptime as 2 hrs 02 min 13 sec.
- Monitor:** Displays CPU, Memory, Classes, and Threads usage. The CPU usage is 0%.
- Heap:** Shows memory usage. The size is 84,070,400 B, and the used memory is 62,392,104 B. The graph shows a fluctuating memory usage over time.
- Classes:** Shows the number of loaded and unloaded classes. Total loaded: 421, Total unloaded: 0.
- Threads:** Shows the number of live and daemon threads. Live: 5, Live peak: 5, Daemon: 4, Total started: 5.

Java - aihw/src/iddfs.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- aihws
 - src
 - (default package)
 - bfs.java
 - bfs2.java
 - bfs3.java
 - iddfs.java
- JRE System Library [JavaSE-1.8]
 - src
 - (default package)
 - ids.java

iddfs.java

```

3 public class iddfs {
4
5     static String[] goalnode = {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","0"}; //Goal State
6
7     public static void main(String args[]){
8         long startTime = System.currentTimeMillis(); //To measure time
9         Stack<String[]> frontier = new Stack<String[]>(); //frontier declaration
10        List<String[]> explored = new LinkedList<String[]>(); //explored declaration
11        // String[] rootnode = {"1","2","3","4","5","6","7","8","9","10","11","13","14","15","12"}; //Initial State 1
12        String[] rootnode = {"1","2","3","4","5","6","7","8","9","10","15","13","14","12","11"}; //Initial State 2
13        // String[] rootnode = {"1","2","3","4","5","0","12","10","9","8","7","11","13","14","15","6"}; //Initial State 3
14        frontier.add(rootnode); //Add initial state to frontier
15        String[] node = rootnode;
16        for(int limit = 1; ;limit++){
17            System.out.println("Node: "+Arrays.toString(node));
18            if(Arrays.equals(node, goalnode)){
19                System.out.println("Solution found: "+Arrays.toString(node));
20                System.exit(0);
21            }
22            else if(limit == 0){
23                System.out.println("Cutoff");
24            }
25        }
26    }
27 }

```

Console

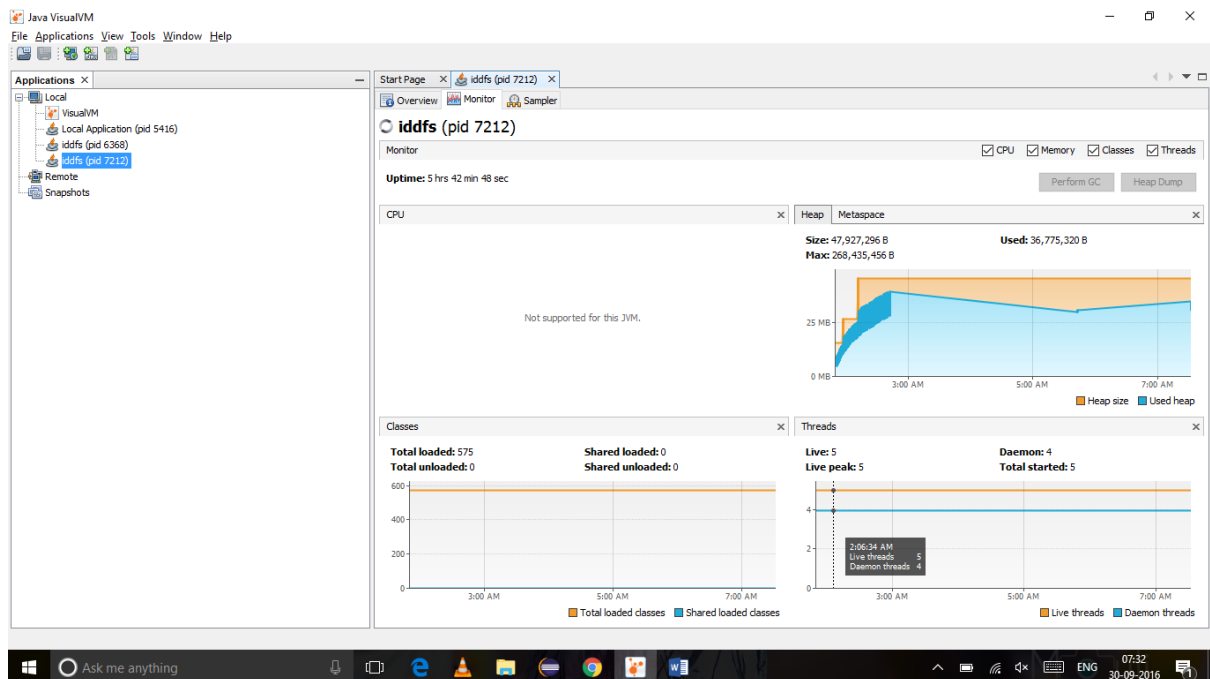
```

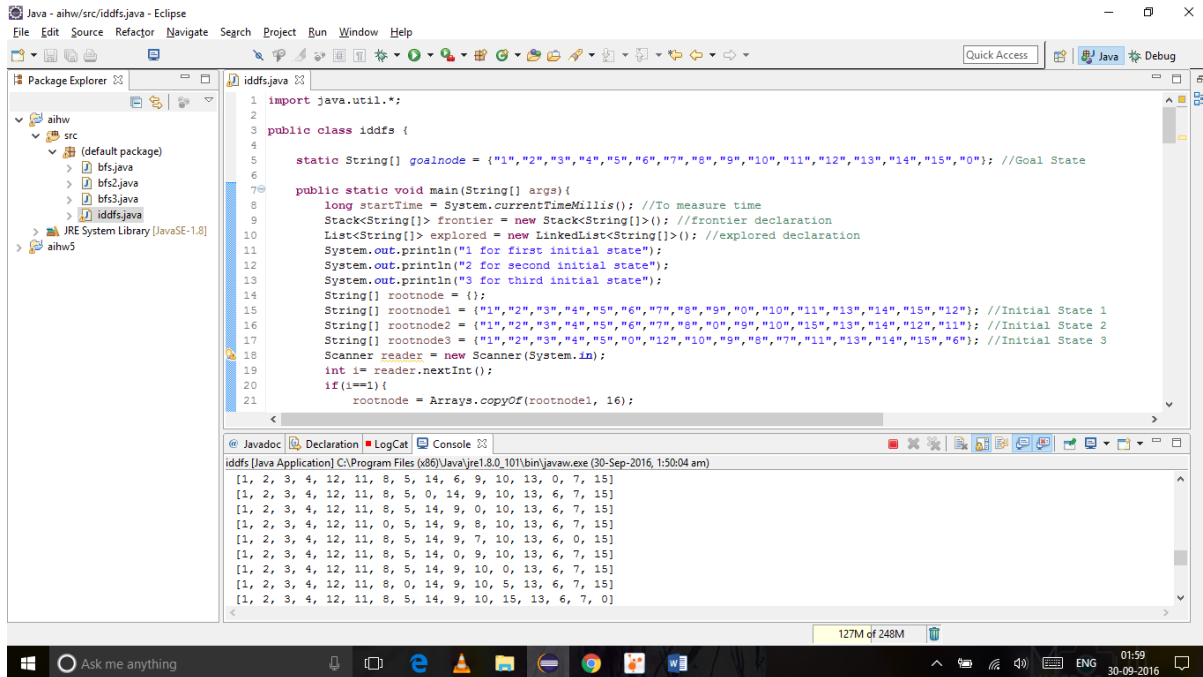
<terminated> iddfs [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (29-Sep-2016, 10:21:43 pm)
[13, 7, 1, 2, 10, 9, 15, 14, 0, 4, 11, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 4, 0, 11, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 0, 15, 14, 4, 9, 11, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 4, 12, 11, 6, 3, 0, 5, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 0, 4, 11, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 4, 11, 0, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 9, 0, 14, 4, 11, 15, 6, 3, 12, 5, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 4, 11, 5, 6, 3, 12, 0, 8]
[13, 7, 1, 2, 10, 9, 15, 14, 4, 0, 11, 6, 3, 12, 5, 8]

```

98M of 248M

Initial State 3:





```
1 import java.util.*;
2
3 public class iddfs {
4
5     static String[] goalnode = {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","0"}; //Goal State
6
7     public static void main(String[] args){
8         long startTime = System.currentTimeMillis(); //To measure time
9         Stack<String[]> frontier = new Stack<String[]>(); //frontier declaration
10        List<String[]> explored = new LinkedList<String[]>(); //explored declaration
11        System.out.println("1 for first initial state");
12        System.out.println("2 for second initial state");
13        System.out.println("3 for third initial state");
14        String[] rootnode = {};
15        String[] rootnode1 = {"1","2","3","4","5","6","7","8","9","10","11","13","14","15","12"}; //Initial State 1
16        String[] rootnode2 = {"1","2","3","4","5","6","7","8","9","10","15","13","14","12","11"}; //Initial State 2
17        String[] rootnode3 = {"1","2","3","4","5","6","7","8","9","10","12","10","9","8","7","11","13","14","15","6"}; //Initial State 3
18        Scanner reader = new Scanner(System.in);
19        int i = reader.nextInt();
20        if(i==1){
21            rootnode = Arrays.copyOf(rootnode1, 16);
22        }
23    }
24 }
```

iddfs [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (30-Sep-2016, 1:50:04 am)

```
[1, 2, 3, 4, 12, 11, 8, 5, 14, 6, 9, 10, 13, 0, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 0, 14, 9, 10, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 14, 9, 0, 10, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 0, 5, 14, 9, 8, 10, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 14, 9, 7, 10, 13, 6, 0, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 14, 0, 9, 10, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 14, 9, 10, 0, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 0, 14, 9, 10, 5, 13, 6, 7, 15]
[1, 2, 3, 4, 12, 11, 8, 5, 14, 9, 10, 15, 13, 6, 7, 0]
```