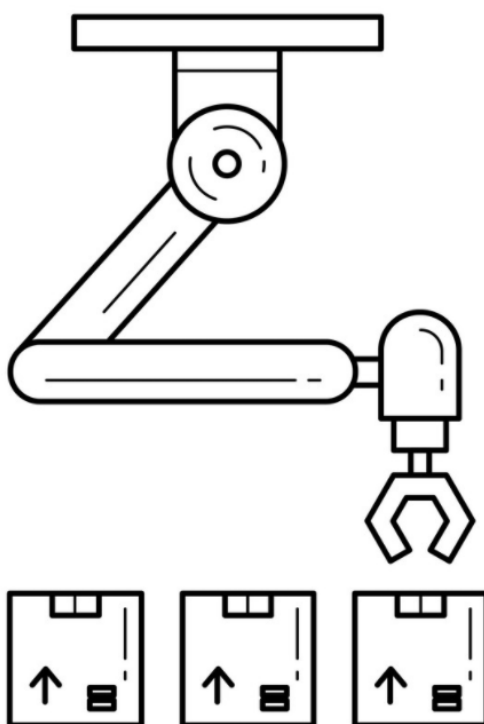




Univerza v Ljubljani
Fakulteta *za računalništvo*
in informatiko

Robotizirano skladišče

Seminarska naloga za premet umetna inteligenca



Avtorja:
Dragan Spasovski
Domen Žukovec

Ljubljana, januar 2021

Povzetek

V poročilu predstaviva drugo seminarsko nalogo za predmet umetna inteligenca. Najprej predstaviva kako sva sprogramirala premikanje in nato prikaževa rezultate, ki sva jih dobila z preiskovalnimi algoritmi DFS (depth-first search), BFS (breadth-first search) in IDDFS / ID (iterative deepening depth-first search).

Ključne besede: preiskovalni algoritmi, DFS, BFS, IDDFS, iskanje rešitve, robotizirano skladišče

Abstract

In the report, we present the second seminar paper for the subject of artificial intelligence. First, we present how we programmed the move function and then, show the results we got with the research algorithms DFS (depth-first search), BFS (breadth-first search) in IDDFS / ID (iterative deepening depth-first search).

Key words: search algorithms, DFS, BFS, IDDFS, goal search, robotic warehouse

Kazalo vsebine

1. Uvod	4
2. Premikanje	4
3. DFS (depth-first search)	5
3.1. Testiranje na primerih	5
4. BFS (breadth-first search)	11
4.1. Testiranje na primerih	11
5. IDDFS (iterative deepening depth-first search).....	14
5.1. Testiranje na primerih	14

1. Uvod

Za drugo seminarsko nalogo smo morali implementirati in uporabiti preiskovalne algoritme na domeni robotiziranega skladišča. Odločila sva se, da bova programirala v programskem jeziku java. Za pomoč pri pisanju algoritmov pa sva uporabila datoteke, ki so objavljene na spletni učilnici. Odločila sva se da bova v najinem programu implementirala algoritme BFS, DSF in ID.

2. Premikanje

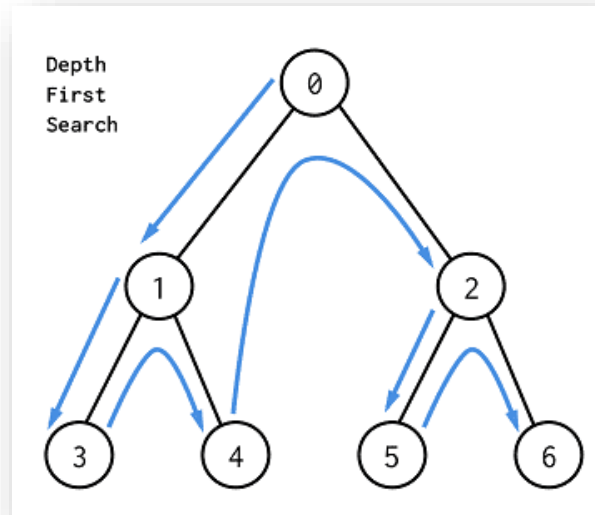
Za premikanje in kreiranje grafa sva naredila razred Node, ki ima parametre n, p, stars, otroci, obstojece, depth, skatle in premik. Razred Node ima funkcije expand, prestavi in isKoncna.

Ker sva se odločila za predstavitev škatel kot matrike ASCII vrednosti črk, funkcija prestavi in expand delata z indexi matrike. V funkciji expand se določita P in R nato se podata v funkcijo prestavi, ki najde prvo škatlo v stolpcu P in jo premakne na prvo dostopno mesto v stolpcu R, nato pogleda če se je ta kombinacija že zgodila in če se ni jo naredi kot novega otroka.

```
public void prestavi(int[][] a, int c, int r)
{
    if (a[0][r] != 0 | c == r) {
        return;
    }
    int skatlaP = 0;
    boolean prazno = true;
    for (int i = 0; i < a.length; i++) {
        if (a[i][c] != 0) {
            prazno = false;
            skatlaP = a[i][c];
            a[i][c] = 0;
            break;
        }
    }
    if (!prazno) {
        prazno = true;
        for (int j = 0; j < a.length; j++) {
            if (a[j][r] != 0) {
                prazno = false;
                a[j - 1][r] = skatlaP;
                break;
            }
        }
        if (prazno) {
            a[a.length - 1][r] = skatlaP;
        }
    }
    boolean obstaja = false;
    for(int[][] x : obstojece)
    {
        if(Arrays.deepEquals(x,a) | Arrays.deepEquals(skatle,a)){obstaja=true;break;}
    }
    if(!obstaja) {
        Node otrok = new Node(a, n, p);
        obstojece.add(otrok.skatle);
        otroci.add(otrok);
        otrok.stars = this;
        otrok.depth = otrok.stars.depth + 1;
        otrok.obstojece = this.obstojece;
        otrok.premik = new int[]{c, r};
    }
}
```

3. DFS (depth-first search)

Iskanje po globini (DFS) je algoritem za prehod drevesnih ali grafičnih podatkovnih struktur. Algoritem se začne pri korenskem vozlišču (lahko si izberemo poljubnega na grafu) in preiskuje, kolikor je mogoče, vzdolž vsake veje.



3.1. Testiranje na primerih

Primer1

```
'A', 'B', 'C'
```



```
'A', 'C',  
'C', 'B',  
'B', 'A',
```

```
Izvedel se je premik: [1, 0]  
*****  
[0, 0, 0]  
[67, 0, 0]  
[66, 65, 0]
```

```
Izvedel se je premik: [1, 0]  
*****  
[65, 0, 0]  
[67, 0, 0]  
[66, 0, 0]
```

```
Stevilo vozlisc: 25  
Potrebovan cas: 46ms  
Globina rezultata: 9  
Max globina: 9  
Stevilo premikov: 9
```

Primer2

```
' ' ' ' ' ' ' '
' ' ' ' ' ' ' '
'B','E',' ' '
'A','C','D'
```



```
' ' ' ' ' ' ' '
' ' ' ' ' ' ' '
'C','D',' ' '
'B','A','E'
```

```
Izvedel se je premik: [1, 0]
*****
[0, 0, 0]
[0, 68, 67]
[66, 65, 69]

Izvedel se je premik: [2, 0]
*****
[0, 0, 0]
[67, 68, 0]
[66, 65, 69]

Stevilo vozlisc: 1080
Potrebovan cas: 138ms
Globina rezultata: 398
Max globina: 400
Stevilo premikov: 398
```

Primer3

```
' ' ' ' ' ' ' '
' ' ' ' ' ' ' '
'E','F',' ' '
'A','C','D','B'
```



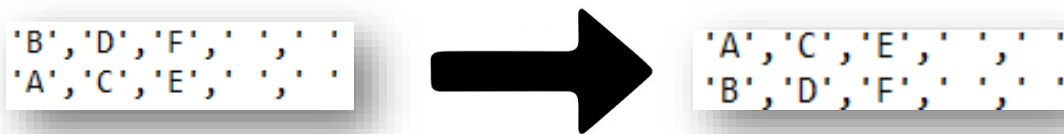
```
' ' ' ' ' ' ' '
' ' ' ' ' ' ' '
'D','E',' ' 'C'
'B','A',' ' 'F'
```

```
Izvedel se je premik: [3, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 68]
[0, 69, 0, 67]
[66, 65, 0, 70]

Izvedel se je premik: [3, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[68, 69, 0, 67]
[66, 65, 0, 70]

Stevilo vozlisc: 48611
Potrebovan cas: 759614ms
Globina rezultata: 6443
Max globina: 12250
Stevilo premikov: 6443
```

Primer4

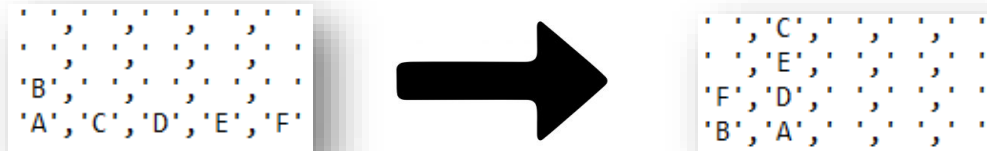


```
Izvedel se je premik: [1, 0]
*****
[65, 0, 69, 0, 0]
[66, 68, 70, 0, 67]

Izvedel se je premik: [4, 1]
*****
[65, 67, 69, 0, 0]
[66, 68, 70, 0, 0]

Stevilo vozlisc: 29923
Potrebovan cas: 57950ms
Globina rezultata: 7071
Max globina: 7071
Stevilo premikov: 7071
```

Primer5

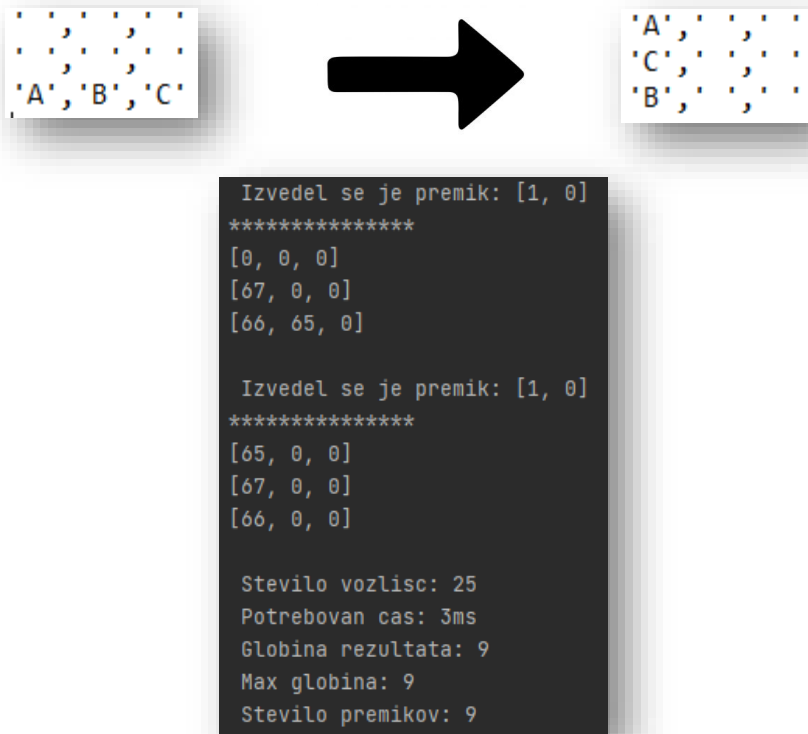


```
Izvedel se je premik: [0, 1]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[0, 68, 0, 0, 0]
[66, 65, 0, 0, 70]

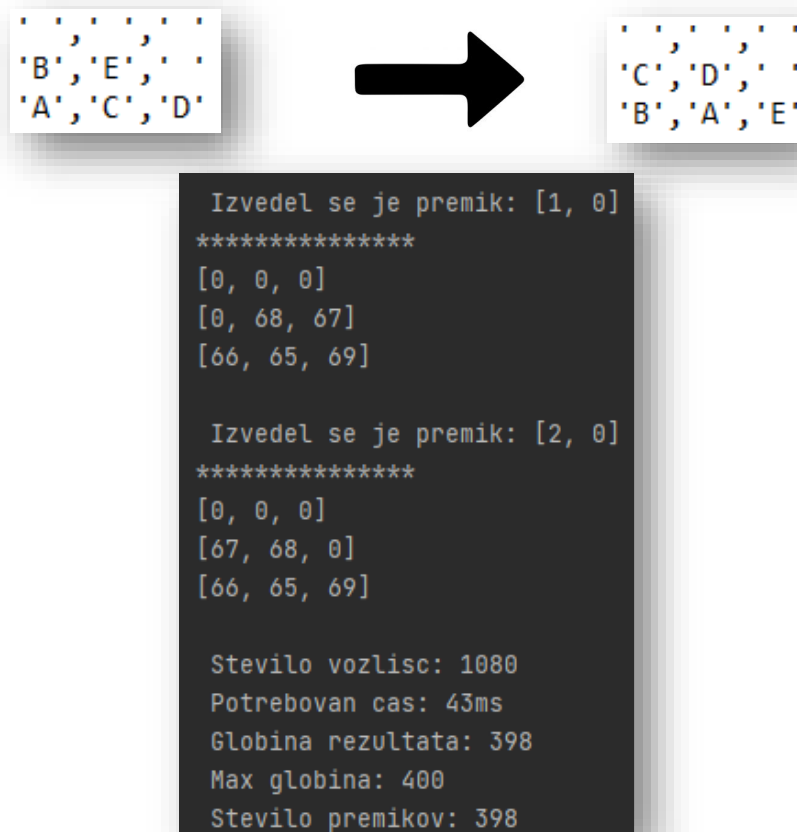
Izvedel se je premik: [4, 0]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[70, 68, 0, 0, 0]
[66, 65, 0, 0, 0]

Stevilo vozlisc: 133200
Potrebovan cas: 7364960ms
Globina rezultata: 29
Max globina: 27183
Stevilo premikov: 29
```

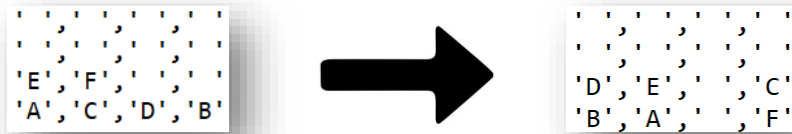
Primer1.2



Primer2.2



Primer 3.2



```
Izvedel se je premik: [3, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 68]
[0, 69, 0, 67]
[66, 65, 0, 70]

Izvedel se je premik: [3, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[68, 69, 0, 67]
[66, 65, 0, 70]

Stevilo vozlisc: 48611
Potrebovan cas: 205524ms
Globina rezultata: 6443
Max globina: 12250
Stevilo premikov: 6443
```

Primer4.2



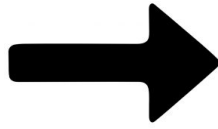
```
Izvedel se je premik: [1, 0]
*****
[65, 0, 69, 0, 0]
[66, 68, 70, 0, 67]

Izvedel se je premik: [4, 1]
*****
[65, 67, 69, 0, 0]
[66, 68, 70, 0, 0]

Stevilo vozlisc: 29923
Potrebovan cas: 17599ms
Globina rezultata: 7071
Max globina: 7071
Stevilo premikov: 7071
```

Primer5.2

```
' ' ',' ',' ',' ',' ',' '
' ' ',' ',' ',' ',' ',' '
' ' ',' ',' ',' ',' ',' '
'B',' ',' ',' ',' ',' '
' ' ',' ',' ',' ',' ',' '
'A','C','D','E','F'
```



```
' ' ','C',' ',' ',' ',' '
' ' ','E',' ',' ',' ',' '
' ' ',' ',' ',' ',' ',' '
'F','D',' ',' ',' ',' '
'B','A',' ',' ',' ',' '
```

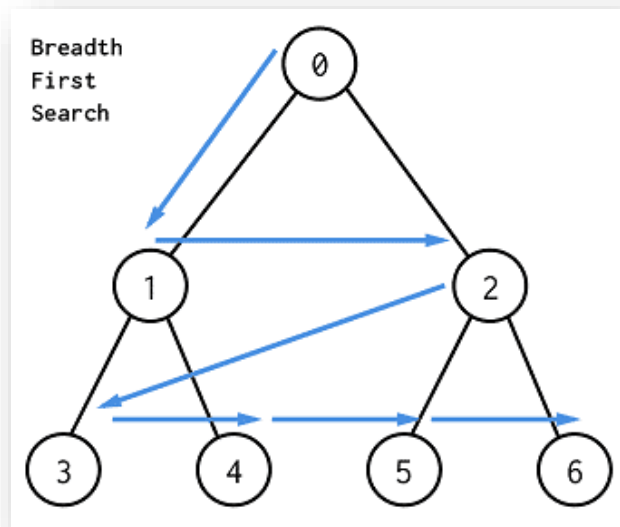
```
Izvedel se je premik: [0, 1]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[0, 68, 0, 0, 0]
[66, 65, 0, 0, 70]

Izvedel se je premik: [4, 0]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[70, 68, 0, 0, 0]
[66, 65, 0, 0, 0]

Stevilo vozlisc: 133200
Potrebovan cas: 7364960ms
Globina rezultata: 29
Max globina: 27183
Stevilo premikov: 29
```

4. BFS (breadth-first search)

Iskanje po širini (BFS) je algoritem za prehod ali iskanje drevesnih ali grafičnih podatkovnih struktur. Začne se pri drevesnem korenu (ali pa katerem koli poljubnem vozlišču grafa) in razišče vsa sosednja vozlišča na določeni globini, preden se premakne na vozlišča v naslednji globini. Uporablja nasprotno strategijo DFS.



4.1. Testiranje na primerih

Primer1

```
.,.,.  
,,.,  
'A','B','C'
```



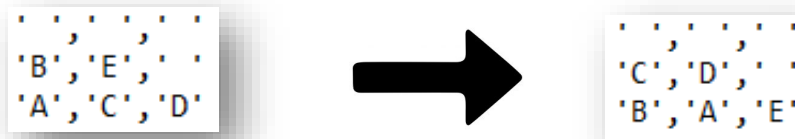
```
'A',',',',',  
'C',',',',',  
'B',',',',',
```

```
Izvedel se je premik: [2, 0]  
*****  
[0, 0, 0]  
[67, 0, 0]  
[66, 65, 0]
```

```
Izvedel se je premik: [1, 0]  
*****  
[65, 0, 0]  
[67, 0, 0]  
[66, 0, 0]
```

```
Stevilo vozlisc: 56  
Potrebovan cas: 46ms  
Globina rezultata: 5  
Max globina: 5  
Stevilo premikov: 5
```

Primer2

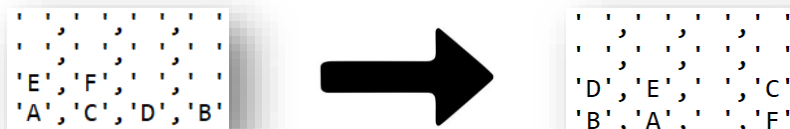


```
Izvedel se je premik: [2, 1]
*****
[69, 0, 0]
[67, 68, 0]
[66, 65, 0]

Izvedel se je premik: [0, 2]
*****
[0, 0, 0]
[67, 68, 0]
[66, 65, 69]

Stevilo vozlisc: 798
Potrebovan cas: 103ms
Globina rezultata: 9
Max globina: 9
Stevilo premikov: 9
```

Primer3



```
Izvedel se je premik: [2, 1]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 69, 0, 67]
[66, 65, 68, 70]

Izvedel se je premik: [2, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[68, 69, 0, 67]
[66, 65, 0, 70]

Stevilo vozlisc: 21828
Potrebovan cas: 56716ms
Globina rezultata: 8
Max globina: 8
Stevilo premikov: 8
```

Primer4

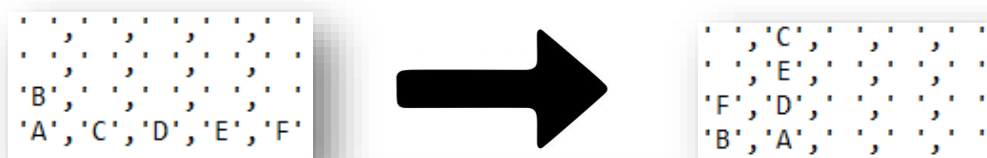


```
Izvedel se je premik: [3, 0]
*****
[0, 67, 69, 0, 0]
[66, 68, 70, 0, 65]

Izvedel se je premik: [4, 0]
*****
[65, 67, 69, 0, 0]
[66, 68, 70, 0, 0]

Stevilo vozlisc: 32400
Potrebovan cas: 224597ms
Globina rezultata: 12
Max globina: 12
Stevilo premikov: 12
```

Primer5



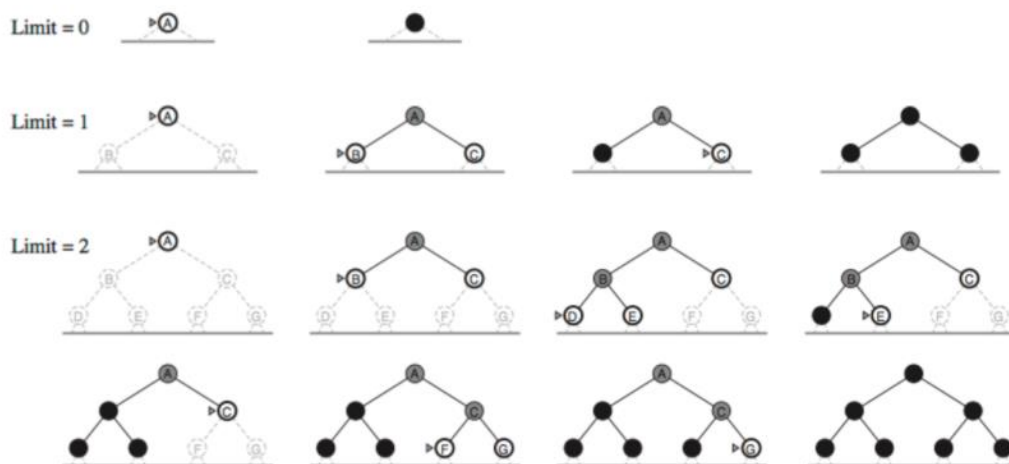
```
Izvedel se je premik: [4, 1]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[0, 68, 0, 0, 0]
[66, 65, 0, 0, 70]

Izvedel se je premik: [4, 0]
*****
[0, 67, 0, 0, 0]
[0, 69, 0, 0, 0]
[70, 68, 0, 0, 0]
[66, 65, 0, 0, 0]

Stevilo vozlisc: 99364
Potrebovan cas: 1895695ms
Globina rezultata: 8
Max globina: 8
Stevilo premikov: 8
```

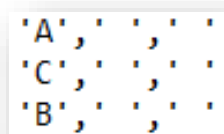
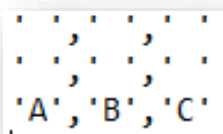
5. IDDFS (iterative deepening depth-first search)

V računalništvu je iterativno poglobljanje iskanja ali natančneje iterativno poglobljanje iskanja po globini (IDS / IDDFS) strategija iskanja po prostoru oz. grafu, pri kateri se različica iskanja po globini izvaja večkrat z vedno večjo globino dokler ne najdemo iskanega rezultata. IDDFS je optimalen tako kot BFS ampak uporablja veliko manj pomnilnika. Pri vsaki ponovitvi obišče vozlišča v iskanem drevesu v istem vrstnem redu kot iskanje po globini (DFS).



5.1. Testiranje na primerih

Primer1



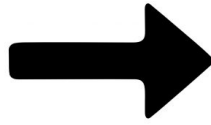
```
Izvedel se je premik: [2, 0]
*****
[0, 0, 0]
[67, 0, 0]
[66, 65, 0]
```

```
Izvedel se je premik: [1, 0]
*****
[65, 0, 0]
[67, 0, 0]
[66, 0, 0]
```

```
Stevilo vozlišc: 56
Potrebovan cas: 51ms
Globina rezultata: 5
Max globina: 5
Stevilo premikov: 5
```

Primer2

'B', 'E',
'A', 'C', 'D'



'C', 'D',
'B', 'A', 'E'

```
Izvedel se je premik: [2, 1]
*****
[69, 0, 0]
[67, 68, 0]
[66, 65, 0]

Izvedel se je premik: [0, 2]
*****
[0, 0, 0]
[67, 68, 0]
[66, 65, 69]

Stevilo vozlisc: 798
Potrebovan cas: 255ms
Globina rezultata: 9
Max globina: 9
Stevilo premikov: 9
```

Primer3

'E', 'F',
'A', 'C', 'D', 'B'



'D', 'E', 'C',
'B', 'A', 'F'

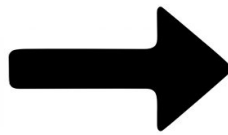
```
Izvedel se je premik: [2, 1]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 69, 0, 67]
[66, 65, 68, 70]

Izvedel se je premik: [2, 0]
*****
[0, 0, 0, 0]
[0, 0, 0, 0]
[68, 69, 0, 67]
[66, 65, 0, 70]

Stevilo vozlisc: 21828
Potrebovan cas: 235814ms
Globina rezultata: 8
Max globina: 8
Stevilo premikov: 8
```

Primer4

```
'B','D','F',' ',' '
'A','C','E',' ',' '
```



```
'A','C','E',' ',' '
'B','D','F',' ',' '
```

```
Izvedel se je premik: [3, 0]
*****
[0, 67, 69, 0, 0]
[66, 68, 70, 0, 65]

Izvedel se je premik: [4, 0]
*****
[65, 67, 69, 0, 0]
[66, 68, 70, 0, 0]

Stevilo vozlisc: 32400
Potrebovan cas: 1623764ms
Globina rezultata: 12
Max globina: 12
Stevilo premikov: 12
```