

Umjetna inteligencija – Laboratorijska vježba 1

UNIZG FER, ak. god. 2013/14.

Zadano: 26.3.2014. Rok predaje: 6.4.2014. do 23.59 sati.

Star Trek problem najkraćeg puta

Kapetan Zvezdane flote Jean-Luc Picard hitno mora doći sa svog broda Enterprise do planeta El-Adrel kako bi spriječio prijeteći sukob s rasom Tamarijanaca. Picard prvo mora s mosta Enterprisea doći do nekog od mjesta za teleportiranje kako bi se mogao teleportirati na planet El-Adrel. Potom s mjesta dolaska na planet, mora doći do dogovorenog mjesta sastanka sa zapovjednikom Tamarijanaca. Kako magnetsko polje planeta ometa normalan rad brodskih sustava, Picard ne može biti siguran na kojem će se točno mjestu na planetu El-Adrel pojaviti. Točnije, svaki od teleportera na Enterpriseu će ga teleportirati na drugo mjesto na planetu El-Adrel. Umjesto teleportacije, Picard se može odlučiti na sigurniji, ali sporiji način putovanja – prijevoz *shuttleom* od Enterprisea do planeta El-Adrel. Pri tome mora od zapovjednog mosta prvo stići do mjesta lansiranja *shuttleova*.

Problem s kojim je Picard suočen može se prikazati 2D rešetkom veličine $N \times N$ (vidi sliku na sljedećoj stranici), i to na način opisan u nastavku. Na polju je označena pozicija mosta broda Enterprise (početno polje, na slici označeno s “P”) te pozicija mjesta sastanka na planetu El-Adrel (ciljno polje, na slici označeno s “C”). Polja na kojima se nalaze teleporteri označena su s “ T_i ” (pri čemu može biti više teleportera s istom oznakom). S teleporterom “ T_i ” Picard se može teleportirati do bilo kojeg polja koje je označeno teleporterom iste oznake. Mjesto lansiranja *shuttleova* na brodu Enterprise označeno je oznakom “ S_S ”, dok je jedino mjesto na kojem *shuttle* može sletjeti na planetu El-Adrel označeno oznakom “ S_L ”. Svako polje sadrži broj koji označava njegovu visinu. Vrijeme prelaska (hodanja) između dva polja određeno je apsolutnom razlikom između visina tih dvaju polja, pri čemu su prijelazi mogući samo između susjednih polja koja dijele jednu stranu (polje ima najviše četiri susjedna polja). Vrijeme teleportiranja jednako je Manhattan udaljenosti između polja s oznakom istog teleportera, dok je vrijeme vožnje *shuttleom* jednako trostrukoj Manhattan udaljenosti između lansirnog (“ S_S ”) i slijetnog polja (“ S_L ”) *shuttlea*. Dolaskom do ruba broda Enterprise, naravno, **nije moguće** hodanjem preći na planet El-Adrel (debela crna vertikalna crta na slici). Početno i ciljno polje, sva polja na kojima se nalaze teleporteri te lansirno i slijetno polje *shuttlea* imaju visinu jednaku 0. Važno je napomenuti da dolaskom do polja na kojem se nalazi teleporter (ili do polja lansiranja *shuttlea*), Picard se može, ali i **ne mora nužno** teleportirati (odnosno poletjeti *shuttleom*), već može (ukoliko procijeni da mu je to vremenski isplativije) preći na neko od susjednih polja.

Vaš zadatak, kao Picardovog časnika za znanost, jest odrediti vremenski najkraći put kojim Picard može doći od zapovjednog mosta Enterprisea do mjesta sastanka na planetu El-Adrel. Napišite program koji pretraživanjem prostora stanja pronalazi optimalno rješenje problema (najkraći put). Optimalno rješenje problema jest redoslijed kretanja (hodanje,

Enterprise						El-Adrel					
7	7	2	19	0	9	8	10	1	9	3	4
3	1	T_3	1	6	0	5	17	12	14	13	1
2	3	8	12	7	S_5	23	16	1	T_3	2	8
P	7	7	8	5	3	11	8	1	7	11	14
4	5	15	8	1	11	3	2	7	8	5	T_2
6	1	3	22	11	7	23	16	12	10	12	4
9	1	4	3	14	13	T_2	6	1	11	S_L	19
12	17	4	1	8	12	2	5	3	14	4	17
11	21	T_1	12	16	T_1	6	14	10	5	13	21
3	9	4	3	4	6	9	2	11	2	11	8
4	9	1	17	3	T_2	10	13	10	T_1	3	C
5	8	21	2	17	11	T_3	1	5	2	1	21

teleportiranje, vožnja *shuttleom*) Picarda po mapi koje ga dovodi od polja “P” do polja “C” u najkraćem mogućem vremenu.

Polja mape jedinstveno su određena dvjema prostornim koordinatama (indeksi redaka i stupaca matrice; brojanje počinje od 1, a ne od 0). Tako, primjerice, na mapi sa slike polje “P” ima koordinate (4,1), polje “C” ima koordinate (11,12), a polje “ S_L ” koordinate (7,11). Optimalno rješenje treba ispisati kao slijed koordinata polja, npr:

(4,1) → (3,1) → (3,2) → (2,2) → (2,3) → (3,10) → (4,10) → (5,10)
 (6,10) → (7,10) → (8,10) → (8,11) → (9,11) → (10,11) → (10,12)
 → (11,12)

Problem je potrebno riješiti:

- Slijepim pretraživanjem, i to uporabom algoritma pretraživanja s jednolikom cijenom;
- Heurističkim pretraživanjem, i to uporabom algoritma A^* , pri čemu je potrebno implementirati sljedeće heuristike:
 - Vrijeme potrebno za dolazak do ciljnog polja sigurno je veće ili jednako od Manhattan udaljenosti između trenutnog polja i ciljnog polja;
 - Vrijeme potrebno za dolazak do ciljnog polja sigurno je veće ili jednako od optimalnog vremena koje bi bilo potrebno na mapi za koju vrijedi: (1) svi teleporteri imaju istu oznaku (tj. sa svakog se teleportera može teleportirati do svake druge pozicije označene teleporterom), (2) dopušteno je samo jedno teleportiranje (i to isključivo s Enterprisea na El-Adrel) te (3) zanemaruju se visine polja te svaki prijelaz hodanja ima trajanje jednako 1.

Pokušajte oblikovati dodatne heuristike koje su potencijalno i bolje od predloženih. Pri tome trebate biti sigurni da je vaša heuristika optimistična budući da je pronalazak optimalnog rješenja problema najvažniji element ove laboratorijske vježbe.

Formati ulaza i izlaza

Program mora biti u stanju učitati konfiguraciju mape iz datoteke. Mapa će biti zadana kao matrica $N \times N$, pri čemu će N uvijek biti paran broj. Granica između Enterprisea i planeta El-Adrel je uvijek na sredini matrice, tj. između stupaca $N/2$ i $N/2 + 1$. Elementi matrice su ili brojevi koji označavaju visinu polja ili pak slova koja označavaju specijalna polja. Za mapu sa slike, matrica u ulaznoj datoteci imala bi sljedeći oblik:

```
7 7 2 19 0 9 8 10 1 9 3 4
3 1 T3 1 6 0 5 17 12 14 13 1
2 3 8 12 7 SS 23 16 1 T3 2 8
P 7 7 8 5 3 11 8 1 7 11 14
4 5 15 8 1 11 3 2 7 8 5 T2
6 1 3 22 11 7 23 16 12 10 12 4
9 1 4 3 14 13 T2 6 1 11 SL 19
12 17 4 1 8 12 2 5 3 14 4 17
11 21 T1 12 16 T1 6 14 10 5 13 21
3 9 4 3 4 6 9 2 11 2 11 8
4 9 1 17 3 T2 10 13 10 T1 3 C
5 8 21 2 17 11 T3 1 5 2 1 21
```

Po pronalasku optimalnog rješenja program treba na standardni izlaz ispisati iznos optimalnog rješenja (najkraće moguće trajanje puta), ukupan broj posjećenih stanja tijekom pretraživanja te ispisati korake koji su doveli do optimalnog rješenja (sve prijelaze između polja, svaki prijelaz u jednome retku). U nastavku su navedena tri primjera koji mogu poslužiti za provjeru ispravnosti vlastitih rješenja (rješenja su dobivena algoritmom A^* uz korištenje druge heuristike; broj otvorenih čvorova vašeg rješenja ne mora se nužno podudarati s brojevima iz primjera).

Primjeri za provjeru

Primjer 1

Ulaz:

```
7 7 2 19 0 9 8 10 1 9 3 4
3 1 T3 1 6 0 5 17 12 14 13 1
2 3 8 12 7 SS 23 16 1 T3 2 8
P 7 7 8 5 3 11 8 1 7 11 14
4 5 15 8 1 11 3 2 7 8 5 T2
6 1 3 22 11 7 23 16 12 10 12 4
9 1 4 3 14 13 T2 6 1 11 SL 19
12 17 4 1 8 12 2 5 3 14 4 17
11 21 T1 12 16 T1 6 14 10 5 13 21
3 9 4 3 4 6 9 2 11 2 11 8
4 9 1 17 3 T2 10 13 10 T1 3 C
5 8 21 2 17 11 T3 1 5 2 1 21
```

Izlaz:

Minimal cost: 31

Opened nodes: 59

Path:

(4,1) ->

(5,1) ->

(5,2) ->

(6,2) ->

(7,2) ->

(7,3) ->

(8,3) ->

(9,3) ->

(11,10) ->

(11,11) ->

(11,12)

Primjer 2

Ulaz:

```
8 3 4 T2 9 1 3 4 7 10 1 6 2 4 3 7
P 1 1 7 3 4 6 SS 8 6 2 4 2 7 4 T1 9
4 T1 8 3 5 T3 8 5 3 8 1 1 2 9 2 3
8 3 T3 8 5 3 7 6 3 T1 8 3 6 9 T3 7
9 1 4 3 7 T1 8 3 6 4 T2 1 9 6 T1 4
8 5 7 4 T4 8 2 9 7 T1 7 2 6 4 9 3
T1 T4 8 2 1 9 8 7 4 8 12 2 SL 9 4 1
8 4 3 7 2 T4 8 9 4 3 2 1 4 5 T1 8
1 2 4 5 9 8 T2 4 5 6 1 2 8 T1 9 4
1 2 8 9 8 4 3 1 1 4 6 4 3 7 5 6
T2 9 8 4 7 T3 7 4 8 2 7 8 9 4 5 5
8 1 3 4 5 8 T4 7 3 8 2 7 4 7 C 5
1 2 9 4 7 8 T3 1 8 T1 8 4 3 5 1 2
9 5 4 8 3 5 2 4 7 5 T1 7 3 9 4 5
3 2 8 9 4 5 3 5 T2 8 4 5 6 8 3 9
1 2 3 4 5 6 7 8 9 T1 7 3 8 2 4 7
```

Izlaz:

Minimal cost: 34

Opened nodes: 166

Path:

(2,1) ->

(2,2) ->

(3,2) ->

(9,14) ->

(10,14) ->

(10,15) ->

(11,15) ->

(12,15)

Primjer 3

Ulaz:

```
7 3 9 8 2 T1 7 4 9 2
P 7 8 4 3 2 6 7 3 9
8 7 3 6 T2 8 6 7 4 8
8 3 6 4 8 2 T1 7 2 8
6 4 7 3 SS 9 7 1 8 2
9 7 T1 2 9 1 4 3 1 3
7 4 9 3 8 2 7 4 SL 9
8 4 3 1 9 3 T2 8 4 3
9 1 2 3 5 3 2 5 7 C
1 8 4 5 7 3 2 8 9 7
```

Izlaz:

Minimal cost: 37

Opened nodes: 91

Path:

```
(2,1) ->
(2,2) ->
(2,3) ->
(2,4) ->
(2,5) ->
(3,5) ->
(8,7) ->
(9,7) ->
(9,8) ->
(9,9) ->
(9,10)
```