

Chiriac Octavian Andrei  
Grupa 233  
Semigrupa 1

## Sfere Cazatoare

Euristica banala: euristica simpla care verifica daca am ajuns la stare finala, caz in care returnam 0 altfel 1.

Euristica admisibila 1: euristica admisibila 1 care pentru fiecare bila din problema gaseste distanta manhattan spre cea mai apropiata iesire. Pentru fiecare bila aceasta distanta va fi calculata si adaugata intr o lista. Se va returna cea mai mare distanta din lista , intrucat euristica va lua in considerare cate blocuri va mai trebui sa strabata bila cea mai de partata de cea mai accesibila poarta

Euristica admisibila 2: euristica admisibila 2 (asemanatoare cu eur1) pentru fiecare bila din problema gaseste distanta manhattan spre cea mai apropiata iesire. Pentru fiecare bila distanata respectiva va fi adunata cu costul necesar pentru a micsora turnul poarta. Acesta suma va fi adaugata intr o lista. Se va returna cea mai mare suma din lista fata de euristica admisibila 1 luand in considerare si ce diferenta este intre turnul pe care se afla bila si turnul cel mai apropiat de aceasta.

Euristica neadmisibila: aceasta euristica are ca scop ajungerea unei bile spre o poarta cat de rapid posibil. Astfel aceasta aduna distantele manhattan catre cele mai apropiate porti fata de pozitiile bilelor. Se va returna suma respectiva la puterea a 3 a. (Aceasta euristica prioritizeaza ca bilele sa se deplaseze catre porti costul drumului fiind neglijat)

Forma apelului: `python main.py folder_input folder_output 2 10`

3	1			
1	2	5	4	5
3	5	4	8	9
2	7	6	6	2
4	1	3	2	4
4	3	6	2	1

sfere

2 2

2 3

iesiri

0 2

4 3

Input_1	lungime	cost	maxim	total	timp
DFS	5	25	22	37272	29.1336
DFI	3	5	27	179	0.06429
BFS	3	5	28	1479	0,21083
A*(banala)	4	5	27	626	0,08676
A*(adm1)	4	5	27	217	0,06881
A*(adm2)	4	5	27	155	0,05134
A*(neadm)	4	5	27	86	0.03091
A*opt(banala)	3	5	27	270	0,08879
A*opt(adm1)	4	5	27	141	0,04487
A*opt(adm2)	4	5	27	117	0,03791
A*opt(neadm)	3	5	27	72	0,00797
IDA*(banala)	4	5	27	3628	0.33524
IDA*(adm1)	4	5	27	699	0.10773
IDA*(adm2)	4	5	27	440	0,06881
IDA*(neadm)	4	11	27	453	0.1325

Forma apelului: `python main.py folder_input folder_output 2 10`

3 1  
 7 8 4 9 7 3 8  
 8 7 11 8 6 1 7  
 3 2 4 2 3 6 102  
 8 6 8 3 12 7 7  
 3 5 5 10 12 5 5  
 4 4 2 1 7 5 10  
 2 3 8 9 4 4 9  
 sfere  
 3 1  
 3 4  
 iesiri  
 5 0  
 6 5

Input_1	lungime	cost	maxim	total	timp
DFS	.	.	.	.	.
DFI	6	24	28	1224100	174.9
BFS	.	.	.	.	.
A*(banala)	.	.	.	.	.
A*(adm1)	6	4	28	1447	0.23044
A*(adm2)	6	4	28	1258	0.21473
A*(neadm)	7	11	28	1323	0.22679
A*opt(banala)	.	.	.	.	.
A*opt(adm1)	6	4	28	413	0.12112
A*opt(adm2)	6	4	28	425	0.0999
A*opt(neadm)	6	4	28	223	0.07333
IDA*(banala)					
IDA*(adm1)	6	4	28	1014	0.16194
IDA*(adm2)	6	4	28	990	0.1623
IDA*(neadm)	.	.	.	.	.

Exemplu pentru neadmisibila:

input1.txt a\*

adm1: cost = 0

```
1)
cost:0
Mat turn:
7 8 4 11 7 3 8
8 7 1 10 6 1 7
3 2 4 9 3 5 102
8 6 9 8 12 4 7
4 3 8 7 3 3 5
4 4 7 6 7 5 10
2 3 8 5 4 4 9
Mat sfere:
. . . # . . .
. . . @ . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . # . . .

6)
turnul (5,2) a scazut cu 2
afis noduri modif
cost:0
Mat turn:
7 8 4 11 7 3 8
8 7 0 10 6 1 7
3 2 4 9 2 5 102
8 6 9 8 9 4 7
4 3 8 7 2 3 5
4 4 5 6 7 5 10
2 3 8 5 4 4 9
Mat sfere:
. . . # . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . % . . .
```

neadm: cost = 4

```
2)
turnul (0,3) a scazut cu 2
afis noduri modif
cost:4
Mat turn:
7 8 4 9 7 3 8
8 7 1 10 6 1 7
3 2 4 9 3 5 102
8 6 9 8 12 4 7
4 3 8 7 3 3 5
4 4 7 6 7 5 10
2 3 8 5 4 4 9
Mat sfere:
. . . % . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . # . . .
```