Problema Blocurilor

Apelare program

Programul poate fi apelat cu urmatoare argumente:

* Fisier de intrare
* Fisier de iesire
* Numar de solutii
* Timeout in milisecunde

Exemplu

python3 main.py input.txt output.txt 3 8

Algoritmi Folositi

* Breadth first
* Depth first: pentru a evita bluclele infinite am introdus un test la generarea succesorilor care verifica daca o stare intermediara a fost folosita sau nu deja in drumul folosit
* Depth first iterativ cu o adancime setata la 20 de iterari
* Uniform cost
* A\* cu urmatoarele euristici:
  + “Euristica banala”: daca ne aflam intr-o stare finala atunci costul pana la starea finala este egal cu 0, altfel este egal cu 1.
  + “Eurstica avansata”: daca ne aflam intr-o stare intermediara calculam suma costul blocurilor deasupra liniei determinate de numarul de blocuri la echilibru in starea finala: nr\_blocuri / nr\_stive

Rezultate

Fisier de intrare cu solutii

Pentru input\_1.txt

Linie de comanda:

*python3 main.py input/input\_1.txt output/output.txt 1 1000000*

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritm | Timp | Lungime drum | Cost |
| BFS | 1.5 ms | 2 | 2 |
| DF | 6.2 ms | 15 | 73 |
| DFI | 0.4 ms | 2 | 2 |
| Uniform cost | 1.4 ms | 2 | 2 |
| A\* euristica banala | 0.4 ms | 2 | 2 |
| A\* euristica avansata | 0.4 ms | 2 | 2 |
| Greedy euristica banala | 1 ms | 2 | 2 |
| Greedy euristica avansata | 1 ms | 2 | 2 |

Pentru input\_2.txt

Linie de comanda:

*python3 main.py input/input\_2.txt output/output.txt 1 1000000*

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritm | Timp | Lungime drum | Cost |
| BFS | 87 ms | 4 | 4 |
| DF | 92 ms | 186 | 841 |
| DFI | 10 ms | 4 | 4 |
| Uniform cost | 255 ms | 4 | 4 |
| A\* euristica banala | 52 ms | 4 | 4 |
| A\* euristica avansata | 6 ms | 4 | 4 |
| Greedy euristica banala | 15 ms | 4 | 4 |
| Greedy euristica avansata | 1 ms | 4 | 4 |

Fisier de intrare fara solutii

Linie de comanda:

*python3 main.py input/fara\_solutii.txt output/output.txt 1 8*

Output linie de comanda:

Configuratie invalida

Fisier de intrare cu timeout

Linie de comanda:

*python3 main.py input/input.txt output/output.txt 3 8*

Output linie de comanda:

Breadth first

Am intrat in timeout.

Depth first

Am intrat in timeout.

Depth first iterativ

Uniform cost

A\* euristica banala

A\* euristica avansata

Fisier de intrare unde starea initiala este finala

Linie de comanda:

*python3 main.py input/starea\_initiala\_este\_finala.txt output/output.txt 3 100*

Output fisier:

Cost: 0

Lungime: 1

Timp: \* ms

1)

[c/1/2] [a/1/0]

[e/10/17] [b/8/0]

[d/10/18] [f/4/0]

[h/3/0]