**Documentatie rezolvare**

**Fisiere de input:**

1. Fara solutii: 6.in
2. Starea initiala e finala: 7.in
3. Nu blocheaza pe niciun algoritm: 5.in si 2.in
4. **Metoda de apelare a programului:**

python main.py {nume folder input} {nume folder output} {nr sol. cautate} {timp timeout}

A screenshot of a computer

Description automatically generated with low confidence

**6. Euristici:** se gasesc sub forma unor functii comentate in cod. Pentru a schimba euristica, se decomenteaza cea dorita si se comenteaza cea actuala.

**-** banala: scorul unei harti este egal cu cati oameni mai sunt de rapit

- euristica neadmisibila: distanta Manhattan pana la un om random, inmultita cu (1 + numarul de oameni de pe harta) – se presupune ca drumul va trece doar prin casute fara casa pe ele.

Aceasta euristica este neadmisibila in urmatoarea situatie: mai avem un om de capturat, iar harta arata astfel:

A picture containing several

Description automatically generated, in “o” aflandu-se urmatorii 2 oameni: v si >

- admisibila 1: reprezinta distanta Manhattan (inmultita cu 1 + nr\_oameni\_harta, ca mai sus), pana la al k-lea cel mai departat om fata de pozitia curenta a OZN-ului

- admisibila 2: distanta pana la cea mai apropiata coloana / linie pe care se misca un om. Stiind ca oamenii se misca ori doar pe o anumita coloana, ori doar pe o anumita linie, OZN-ul va trebui sa se deplaseze pana la cel putin o astfel de linie/coloana, asa ca o consideram pe cea mai apropiata ca sa ne asiguram ca va fi admisibila euristica.

Asadar, distanta pana la solutie este egala cu 4. Cu toate acestea, daca omul random ales este cel aflat in coltul din stanga sus, atunci distanta estimate de euristica este considerabil mai mare.

**10. Validari**

- Verificarea corectitudinii datelor de intrare: in clasa Human, la initializare, se verifica faptul ca nu e nicio cladire in traseul unui om; tot la initializare, o verificare e dedicata determinarii daca intervalul are capete comune sau nu e orientat pe linie si coloana; functia isSeen din clasa Node este folosita pt a verifica faptul ca pozitia initiala a navei nu este in raza vizuala a unui om

- Realizarea faptului ca problema nu are solutii din starea initiala: singura situatie in care se poate determina acest lucru este atunci cand nu se poate efectua nicio mutare din pozitia initiala, iar acest caz este tratat prin verificarea daca un nod successor reprezinta o stare invalida inca dinaintea adaugarii sale in lista de succesori, in functia isSeen.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_ CERINTA**

**Extratereștrii sunt printre noi**

**Context**

O navă extraterestră dorește să răpească niște oameni pentru a culege informații despre specia noastră și ne cere ajutorul pentru găsirea unui algoritm. Oamenii sunt dispuși într-un oraș dreptunghiular precum harta de mai jos. Locurile de mers (străzile) sunt marcate cu simbolul "." și locurile ocupate de clădiri sunt marcate cu "#".

selecteaza textul...######........##  
............####...  
..####..#.###......  
..#####..#.........  
.........##..#####.  
###.#..............  
###.#####....#####.  
..###............#.  
.......#######...#.

Fiecare om se plimbă în treburile sale zilnice doar pe un segment (îl vom nota cu AB) (dus-întors) care se află pe linie și sau coloană și cuprinde doar spații libere. Pentru fiecare om se vor da punctele A și B ale segmentului, un punct fiind definit prin linia și coloana pe care se află. "Segmentul" trebuie să treacă doar prin spații libere, nu prin clădiri.

Nava extraterestră nu se poate deplasa dacă ajunge în raza vizuală a altui om, fiindcă altfel omenirea va afla despre existența extratereștrilor și extratereștrii nu vor mai putea să ne studieze în pace.

**Stări și tranziții.**

O stare reprezintă locațiile și direcția oamenilor și poziția navei la un moment dat. O tranziție e o deplasare a navei și a oamenilor. Mutările se fac în următoarea ordine: se mută toți oamenii care au mai rămas pe hartă) pe direcția curentă de deplasare în ordinea în care au fost definiți în fișier. Dacă un om a ajuns în capătul segmentului de deplasare, își schimbă direcția. Schimbarea de direcție durează cât o mutare, cu alte cuvinte, când omul ajunge în capăt la următoarea schimbare de stare, nu face un pas în altă celulă ci doar "își întoarce privirea". Abia la următorul moment de timp va păși în noua direcție.

După ce au fost poziționați oamenii, se mută nava (mutarea navei e obligatorie; dacă nu se poate conform condițiilor problemei, înseamnă că nu e o stare validă), deci se va ține cont de orientarea (și privirea) oamenilor după mutarea lor. Deoarece după ce a trecut nava printr-un loc oamenii devin mai vigilenți și instalează camere de luat vederi, nava nu are voie să revină într-o locație în care a mai fost, fiindca asta ar ajuta la confirmarea existenței ei. De asemenea, nava nu se poate deplasa în raza vizuală a unui om. Ca să fie în raza sa vizuală, nava trebuie să fie pe linia sau coloana de deplasare a omului, în direcția sa de deplasare și fără să existe obstacole sau alți oameni **între** ei (nava se poate afla și pe clădire și tot să fie vizibilă dacă e pe o căsuță din margine atâta timp cât respectă condiția zisă anterior).

Pentru ca nava să răpească un om trebuie să se poziționeze exact pe căsuța în care se află acesta.

În traseele lor, se poate întâmpla ca doi sau mai mulți oameni să se întâlnească în aceeași celulă. In acel moment oamenii se pun pe vorbit și nu se mai uită pe direcția de mers, deci nava poate trece neobservată (la următoarea mutare, oamenii care "vorbeau" în aceeași celulă, își continuă traseul ca și până atunci). Dacă nava ajunge într-o celulă cu mai mulți oameni, îi răpește pe toți.

**Cost**

Costul se calculează astfel:

* la fiecare mutare a navei în spațiul liber(stradă) costul este dat de 1+(numărul de oameni de pe hartă).
* O mutare într-o celulă cu obstacol are costul egal cu 1+(de 2 ori numarul de oameni de pe hartă)

**Stare finală.**

Nava își termină misiunea după ce răpește K oameni.

**Fisierul de intrare**

*Programul citește din fișier:*

* *poziția inițială a navei dată ca o pereche de numere reprezentând linia și coloana.*
* *numărul K de oameni de răpit*
* *harta*
* *cuvântul "oameni" care apare imediat după hartă și indică faptul că începe enumerarea oamenilor*
* *pentru fiecare om va exista câte o linie cu 4 numere, reprezentând linia și coloana poziției de start, linia și coloana celuilalt capăt al segmentului de deplasare. Se va verifica faptul că segmentul e corect (nu are în interior decât spații libere si e orientat doar pe linie sau coloană. De asemenea, nu e voie ca segmentele să aibă capete comune (vor fi doar celule distincte)*

Se va verifica de asemenea ca poziția inițială a navei să nu fie în raza vizuală a vreunui om.

Exemplu de fișier de intrare:

selecteaza textul2 3  
5  
...######........##  
............####...  
..####..#.###......  
..#####..#.........  
.........##..#####.  
###.#..............  
###.#####....#####.  
..###............#.  
.......#######...#.  
oameni  
1 0 1 7  
8 14 8 16  
7 9 5 9  
7 16 7 10  
4 0 4 8  
6 3 5 3  
4 6 4 1  
1 18 1 17  
3 18 7 18  
5 16 5 18  
0 0 0 3  
1 7 5 7  
8 0 8 5  
5 6 5 12

**Fișier output.**

În fișierul de output se va afișa pentru fiecare nod din drumul soluție, starea curentă a hărții în formatul cerut mai jos, poziția fiecărui om care mai e pe hartă, costul total al mutărilor navei de până atunci (inclusiv ultima mutare), o matrice care arată pozițiile în care a mai fost nava (va fi harta orașului, fără oameni, iar pozițiile anterioare ale navei vor fi marcate cu "@"), numărul de oameni capturați.

Afișarea starii curente a hărții se va face cu următoarele notații:

* Simbolul "." pentru loc liber (celulă de stradă, fără om în acea locație)
* Simbolul "#" pentru celulă ocupată de clădire, în cazul în care nava nu e în acea celulă
* Simbolurile "<" pentru om orientat spre stânga, ">" pentru om orientat spre dreapta, "^" pentru om orientat în sus, "v" pentru om orientat în jos.
* Simbolul "o" pentru un grup de doi sau mai mulți oameni care s-au întâlnit în aceeași celulă
* Simbolul "@" pentru navă - acest simbol are prioritate față de simbolurile de mai sus (dacă nava e într-o celulă, întotdeauna se va afișa în poziția corespunzătoare ei simbolul navei

De exemplu, pentru datele de intrare de mai sus, harta inițială din fișierul de ieșire ar fi:

selecteaza textulv..######........##  
>......v....####..<  
..@###..#.###......  
..#####..#........v  
>.....<..##..#####.  
###.#.>.........>..  
###^#####....#####.  
..###....^......<#.  
>......#######>..#.

Nava, de exemplu, nu ar putea să o ia în sus pentru că ar fi văzută de omul de pe linia 1 (care era în poziția 1,0 și avansează acum în 1,1). Un exemplu bun de succesor ar fi următorul, când nava a coborât cu o poziție.

selecteaza textul...######........##  
v>..........####.<.  
..####.v#.###......  
..@####..#.........  
.>...<...##..#####v  
###^#..>.........>.  
###.#####^...#####.  
..###..........<.#.  
.>.....#######.>.#.

Un nou succesor ar putea fi un pas la dreapta, deoarece omulețul de pe coloana respectivă a ajuns în capătul segmentul și se va întoarce așa cum se vede mai jos, și stim că întâi se calculează mutările oamenilor și apoi ale navei. De asemenea observați pe linia 5, coloana 18 întâlnirea celor 2 oameni care e simbolizată prin litera *o*.

selecteaza textul...######........##  
..>.........####.>.  
v.####..#.###......  
..#@###v.#.........  
..>.<....##..#####.  
###v#...>^........o  
###.#####....#####.  
..###.........<..#.  
..>....#######..>#.

Cei doi oameni de pe linia 4 care sunt față în față la următorul pas se vor întâlni în aceeași căsuță, deci nava poate coborî și să îi răpească pe amândoi cu o singură mutare. Omul din dreapta navei nu vede nava fiindcă nu e orientat spre ea; idem cel de pe coloană în jos.

selecteaza textul...######........##  
...>........####..>  
..####..#.###......  
v.#####..#.........  
...@...v.##..#####.  
###.#....o........<  
###v#####....#####v  
..###........<...#.  
...>...#######..<#.

Harta căsuțelor parcurse de navă, pentru starea de mai sus este:

selecteaza textul...######........##  
............####...  
..@###..#.###......  
..@@###..#.........  
...@.....##..#####.  
###.#..............  
###.#####....#####.  
..###............#.  
.......#######...#.