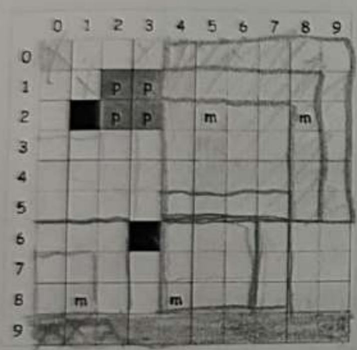
Problema 2

Fie o tablă pătratică de latură N>2 pe care se află un pătrat "plimbător" de latură L<N, NO celule obstacol și alte NM celule mâncare. La un pas pătratul se poate muta doar câte o poziție pe orizontală (cu cost C=1) sau pe verticală (cost C=2) atâta timp cât în urma mutării toate celulele pătratului se află pe tablă și nu se suprapun cu celule obstacol.

Pătratul nostru colectează x celule cu mâncare dacă, în urma mutării, oricare din celulele sale se suprapun cu acestea. Dimensiunea pătratului crește cu x pe direcția dreapta-sus, iar celulele dispar de pe tabla de joc. **Atentie, în cadrul aceleiași mutări**, dacă, în urma creșterii, pătratul se suprapune cu noi celule mâncare, operația se repetă până nu mai există astfel de celule. O mutare **nu** este validă dacă, după finalizarea unui lanț de creșteri, pătratul s-ar suprapune cu un obstacol sau ar ieși de pe tabla de joc.

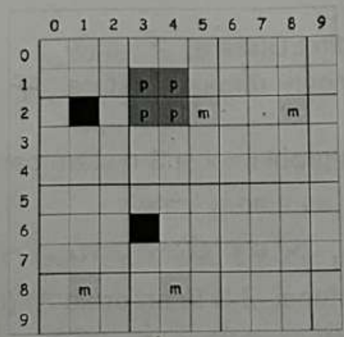
Scopul jocului este să colectăm toate celulele mâncare de pe tablă cu un cost minim.

Starea 1, inițială. N=10 L=2



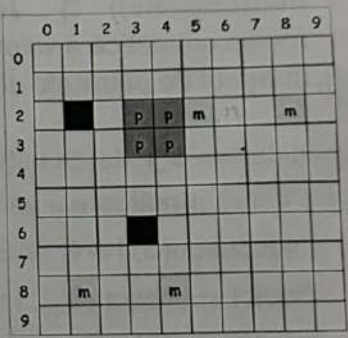
Eh

Starea 2. Mutare la dreapta; C=1

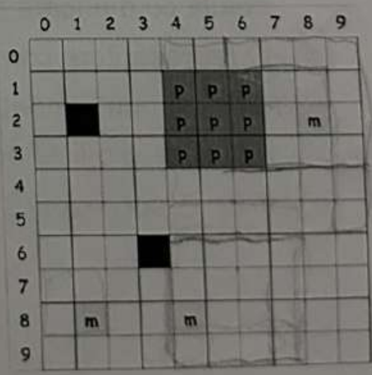


8-16+1+4

Starea 3. Mutare în jos; C=2



Starea 4. Mutare în dreapta; C=2, L devine 3



Notații/convenții:

- Fie M mulțime, **|M|** = cardinalul mulțimii M.
- Fie N număr, **|N|**= valoarea absolută a lui N
- pentru $X \in \mathbb{R}$, **[X] = parte întreagă din X**
- **dist_MH((i1,j1), (i2,j2))** = $|i1 - i2| + |j1 - j2|$; distanța Manhattan între G[i1][j1] și G[i2][j2]
- **dist_E (i1,j1), (i2,j2))** = $\sqrt{(i1 - i2)^2 + (j1 - j2)^2}$; distanța Euclidiană între G[i1][j1] și G[i2][j2]
- **succesori(S)** = mulțimea succesorilor stării S;
- **solutii(S)** = mulțimea drumurilor-**soluție**
- **adâncimea unui nod N în arbore** = numărul de muchii de pe drumul de la rădăcină până la N

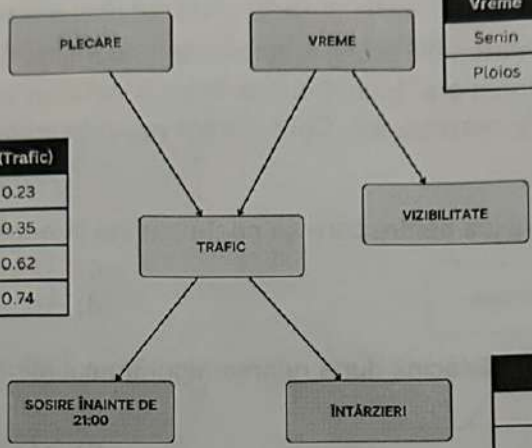
Subiectul 3

Riscurile de a merge cu trenul par să crească cu fiecare minut care trece. Eroul nostru ar fi ieșit deja din Constanța dacă pleca atunci când a început să calculeze. Pentru ca timpul să nu fi fost pierdut, decide să calculeze care din următoarele 2 trenuri programate îi asigură șanse mai mari de a ajunge la București înainte de ora 21:00, conform graficului de mai jos:

Notă: Valorile sunt ~~trunchiate la~~ 2 zecimale.
Notă 2: Considerăm Plecare și Vreme variabile dovezi

Plecare	P (Plecare)
18:00	0.2
18:30	0.8

Vreme	P (Vreme)
Senin	0.7
Ploios	0.3



Plecare	Vreme	P (Trafic)
18:00	Senin	0.23
18:00	Ploios	0.35
18:30	Senin	0.62
18:30	Ploios	0.74

Vizibilitate	P (Vizibilitate)
Da	0.46
Nu	0.62

Trafic	P (Sosire)
Da	0.78
Nu	0.56

Trafic	P (Întârzieri)
Da	0.92
Nu	0.13

9. Alege afirmațiile adevărate:
   a) P (Trafic = False | Plecare = 18:00) = 0.58 F =0.71
   b) P (Trafic = False | Vreme = Ploios) = 0.91 F 0.455
   c) P (Trafic = True | Plecare = 18:00) = 0.58 F 0.29
   d) P (Trafic = True | Vreme = Ploios) = 0.54 A 0.545

10. Bifează toate afirmațiile corecte:
   a) $E^{+}_{Întârzieri}$ = {Plecare, Vreme, Trafic, Vizibilitate} corect
   b) E^{+}_{Vreme} = {Vizibilitate, Trafic, Sosire, Întârzieri} eff ca la c)
   c) E^{+}_{Vreme} = {Vizibilitate, Trafic} nu are ascendenți
   d) $E^{+}_{Întârzieri}$ = {Trafic} e a) corect

11. Care dintre următoarele afirmații este adevărată?
   a) Drumul Vreme->Trafic-> Întârzieri e blocat conditionat de mulțimea {Vizibilitate} F
   b) Drumul Plecare->Trafic-> Sosire blocat conditionat de mulțimea {Trafic} A
   c) Mulțimea {Sosire} d-separă mulțimea {Plecare, Vizibilitate} de mulțimea {Întârzieri, Sosire} F nu sunt disjuncte
   d) Mulțimea {Trafic, Vizibilitate} d-separă mulțimea {Plecare, Vreme} de mulțimea {Sosire, Întârzieri}

12. Care e probabilitatea *a priori* ca Sebastian să ajungă înainte de ora 21:00, dacă pleacă la ora 18:00, iar vremea este senină?
   a) 0.23 b) 0.61 c) 0.69 d) 0.63

4. Ce noduri vor fi în lista open după ce expandăm Drajna Nouă?
   a) Lehliu Gara
   b) Hârșova
   c) Cernavodă
   d) Slobozia

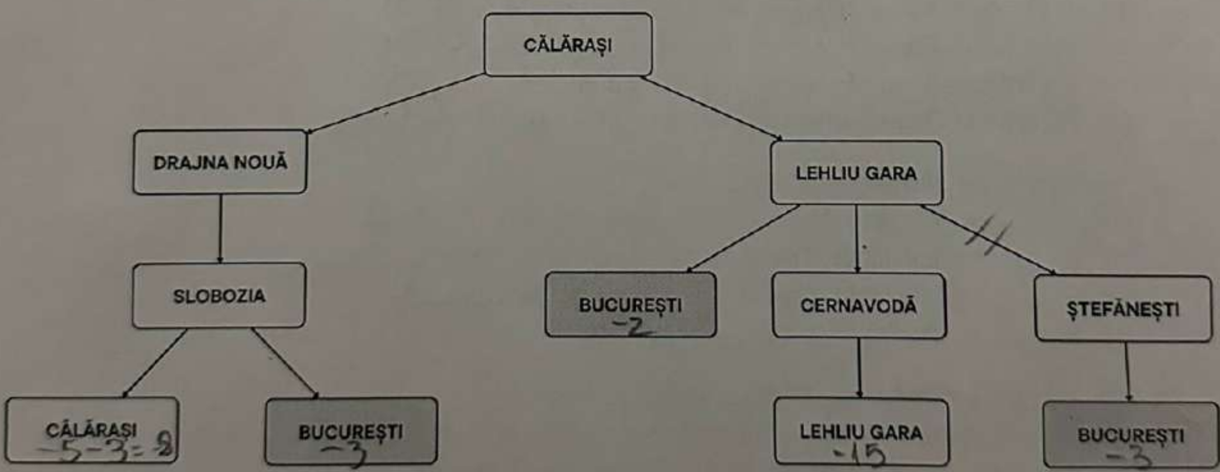
Subiectul 2

Sebastian a luat în derâdere mersul trenurilor așa că CFR călători i-a pus gând rău. Pentru fiecare drum ales de Sebastian, CFR poate amâna pe perioadă nedeterminată toate trenurile de la destinație mai puțin unul, obligându-l astfel să aleagă un anumit drum. Considerăm estimarea unui nod $U_{oraș} = -\ durata\ de\ pe\ mersul\ trenurilor\ -\ trenuri$, unde durata de pe mersul trenurilor este estimarea din graful anterior, iar numărul de trenuri schimbate este adâncimea la care ne aflăm în graf (numărăm de la 0). Sebastian vrea să minimizeze numărul de schimburi, iar CFR să le maximizeze. Considerăm varianta arborescentă a grafului de pe pagina anterioară.

5. Alege orașele de mai jos pentru care se poate ajunge la euristica dată (pe orice drum):
   a) $U_{Cernavodă} = -14$
   b) $U_{Lehliu\ Gara} = -16$
   c) $U_{Hârșova} = -1$
   d) $U_{Drajna\ Nouă} = -6$

6. Cât va fi valoarea din rădăcină după rularea algoritmului MinMax adâncime 2?
   a) -14
   b) -11
   c) -20
   d) -22

Fie următorul subarbore, pe care aplicăm Alpha-Beta pruning (considerăm rădăcina nod MAX):



7. Alege ce noduri vor fi retezate:
   a) Slobozia
   b) Ștefănești
   c) Călărași
   d) Cernavodă

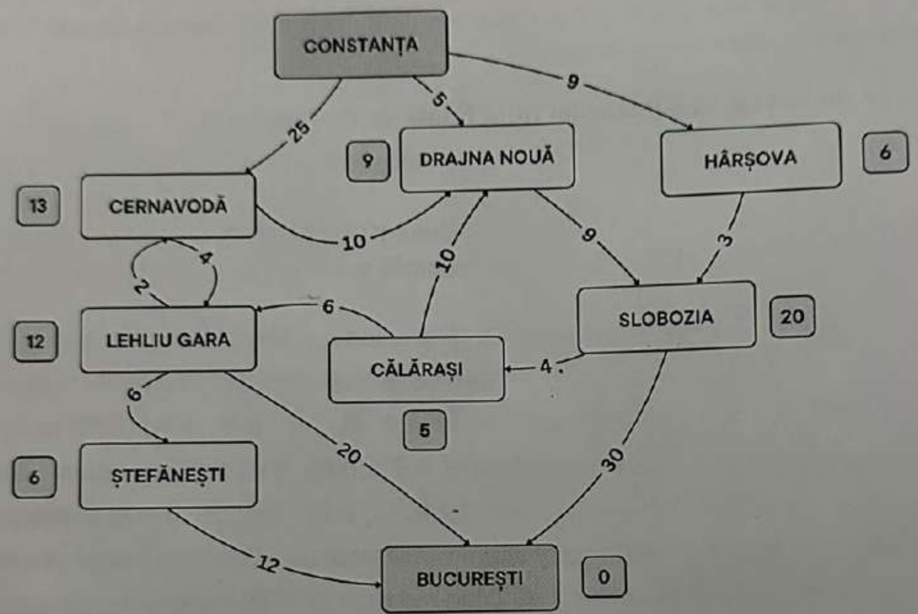
8. Bifează tot ce observi după rularea algoritmului:
   a) Alpha este -3 în Cernavodă A
   b) Beta este -3 în Slobozia F $\beta = \infty$, $\alpha = -3$
   c) Alpha este -8 în Drajna Nouă A $\alpha = -\infty$, $\beta = -3$
   d) Beta este -8 în Drajna Nouă F, $\alpha = -\infty$, $\beta = -3$

Problema 1

Sebastian a fost la mare de ziua lui, dar acum trebuie să se întoarcă la București ca să dea un examen la facultate. El își dorește să ajungă cât mai curând fiindcă a învățat deja că cu cât prinde mai mult timp de somn înainte de examen cu atât va rezolva subiectele mai bine. Ajutați-l pe Sebastian!

Subiectul 1

Presupunem drumul trenurilor ca în imagine. Orașele sunt noduri în graf, muchiile reprezintă durata drumului direct cu trenul între ele, iar euristicile sunt duratele afișate pe mersul trenurilor. Vom aplica algoritmul A* pentru a stabili cel mai bun drum:



1. Selectează nodurile pentru care costul estimat f este corect:
   a) f(Hârșova) = 9 F =15
   b) f(Slobozia) = 18 F =32
   c) f(Cernavodă) = 38 F=37 ? poate fi și 38
   d) f(Drajna Nouă) = 9 =14

2. Dacă mersul trenurilor ar actualiza o euristică, ce variante de mai jos ar păstra regula de admisibilitate?
   a) Lehliu Gara -> 20 F C=18
   b) Ștefănești -> 22 F C=12
   c) Slobozia -> 30 F C=28
   d) Ștefănești -> 6 A

3. Care este primul lanț soluție determinat de algoritmul A*?
   a) Constanța -> Hârșova -> Slobozia -> București 42
   b) Constanța -> Drajna Nouă -> Slobozia -> București 44
   c) Constanța -> Hârșova -> Slobozia -> Călărași -> Lehliu Gara -> Ștefănești -> București 40
   d) Constanța -> Drajna Nouă -> Slobozia -> Călărași -> Lehliu Gara -> Ștefănești -> București 42

13. Numim ST_0 (N, NO, NM, L) mulțimea tuturor stărilor inițiale care se pot crea pentru N, NO, NM, L fixate. Care dintre următoarele variante sunt adevărate?

a) ∀ S ∈ ST_0(10, 1, 1, 2), | solutii(S) | > 0 F, ∃ o stare nu poate mânca 100%
b) ∀ N>2, ∃ S ∈ ST_0(N, N*(N+1)/2, [N/2], 1) cu | solutii(S) | > 0 ? F m=3 nu => 100% fals
c) ∃ S ∈ ST_0(N, $N^2 - (L + NM)^2$, NM, 2), ∀ NM, cu 0 < NM < N-L, astfel încât |soluții (S)| > 0 F 100%
d) Există o coloană (în starea inițială din imaginea dată în enunț) pentru care, orice loc liber am alege pentru a plasa un obstacol, problema nu mai are soluție F, oriunde am pune x pe ultima linie, e ok (i.e. o căsuță din fiecare coloană)
e) Pentru starea inițială dată în enunț, oriunde am plasa un obstacol pe un loc liber de pe diagonală principală, problema va ajunge să nu aibă soluție F, dacă pun la 0,0 are sol

14. Fie LD(stare) = { dist_MH((i1, j1), (i2, j2)) | G[i1][j1] = 'm', G[i2][j2] = 'p' }, LE(stare) = { dist_E((i1,j1), (i2,j2)) | G[i1][j1] = 'm', G[i2][j2] = 'p' } și funcția euristică ĥ, unde ĥ(stare_scop) = 0. Estimația este admisibilă dacă pentru orice altă stare aplicăm formula:

a) min(LD(stare))
b) media aritmetică a valorilor din LE(stare) F 100%
c) max(LE(stare)) F 100%
d) media aritmetică a valorilor din LD(stare) F 100%
e) numărul de pătrățele de mâncare rămase în stare F 100%

15. Considerăm **S_I** starea inițială a problemei și **S** o stare oarecare a problemei, din arborele A*. Care dintre următoarele variante sunt adevărate?

a) | succesori(S) | ≤ 4, ∀ S. [redacted]
b) Dacă | succesori(S_I) | ≠ 0, atunci, ∄ S în arborele A* (cu S_I rădăcină), având |succesori(S)| = 0. ba da, stările finale => F
c) O reprezentare completă (cu toate informațiile necesare) a lui S poate fi formată dintr-o matrice de dimensiune N*N în care am notat cu 'l' un loc liber, cu 'o' un obstacol, cu 'm' o celulă cu mâncare și cu 'p' celulele ocupate de pătrat. A
d) O reprezentare completă (cu toate informațiile necesare) a lui S poate fi compusă dintr-un număr conținând dimensiunea pătratului, o listă cu coordonatele celulelor cu mâncare și o listă cu coordonatele obstacolelor. F nu știm unde e pătratul
e) ∀ Sc ∈ succesori(S), unde | succesori(S) | > 0, avem că g(Sc) - g(S) ≤ 2 A

că cost max = 2
≤2 diferența de la o st la alta