NUME, GRUPĂ:____

DATĂ:____________________

Examen la disciplina Inteligență Artificială,
partea de "Căutarea și Reprezentarea Cunoștințelor"
- Varianta 5 -

1. Să presupunem că avem o problemă de căutare P pentru care vrem să aplicăm algoritmul A*. Graful asociat problemei este un graf orientat oarecare G = (N, A) cu arce ponderate (unde N este mulțimea nodurilor și A este mulțimea arcelor). Ponderile sunt **numere raționale pozitive nenule**. Care dintre următoarele estimații ĥ sunt admisibile **pentru orice astfel de graf**?

- ĥ(n) = min({ct| ct cost al arcului a, unde a este un arc care intră în nodul n}) dacă *n* nu este nod scop și ĥ(n) = 0 dacă *n* este nod scop
- ĥ(n) = min({ct| ct cost al arcului a, unde a este un arc care iese din nodul n}) dacă *n* nu este nod scop și ĥ(n) = 0 dacă *n* este nod scop
- ĥ(n) = 1 dacă *n* nu este nod scop, ĥ(n) = 0 dacă n este nod scop
- Pentru orice nod *n*, ĥ(n) = min({ct| ct cost al arcului a, a ∈ A}) (altfel spus, ct este costul minim pe un arc din graf)
- ĥ(n) = 0 oricare ar fi nodul *n* din graf

2. Care dintre afirmațiile de mai jos sunt adevărate?

- Metoda înlănțuirii înapoi presupune o căutare a răspunsului pornind de la date în încercarea de a găsi scopul.
- Un sistem expert, pe baza informațiilor primite, oferă întotdeauna răspunsuri în totalitate certe.
- Un sistem expert trebuie să poată genera explicația (demonstrația) pentru un răspuns dat.
- Un sistem expert poate funcționa și fără o bază de cunoștințe.
- Un sistem expert este întotdeauna proiectat pentru a răspunde la întrebări din orice domeniu.

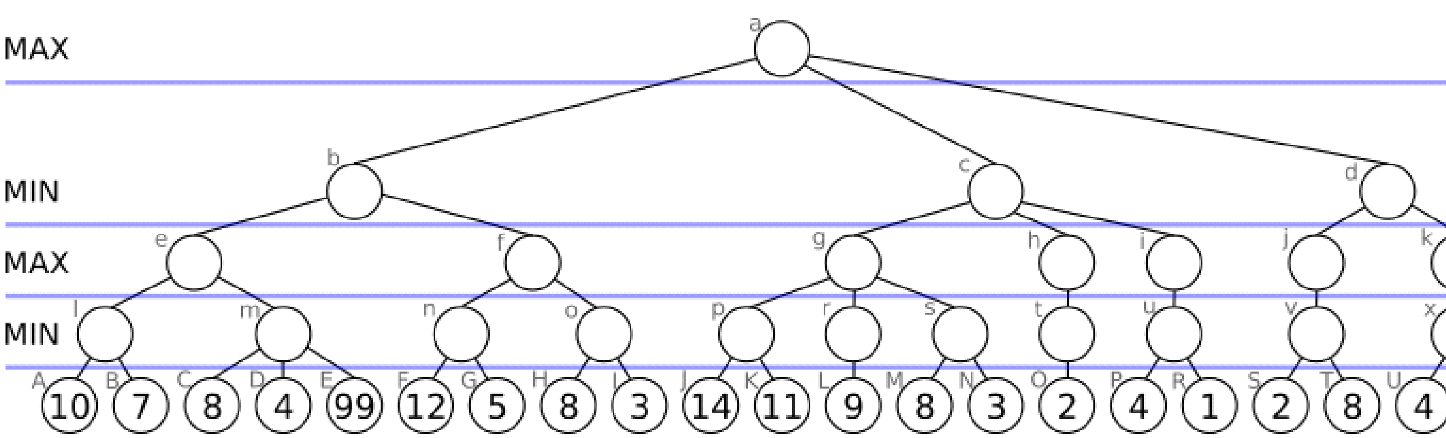
3. Pentru o problemă oarecare (deci afirmația trebuie să fie adevărată pentru orice problemă de căutare cu costuri strict pozitive pe tranziții), care dintre formulele următoare ar obține din estimația admisibilă ĥ(nod), oricare ar fi aceasta (adică formula trebuie să fie adevărată **pentru orice ĥ** nu pentru cazuri particulare de ĥ), o nouă estimație $\hat{h}_1$(nod) **în mod cert** **neadmisibilă?** **Observație: în caz că nu se specifică nimic despre nodul dat ca parametru, înseamnă că se aplică formula pentru orice nod din graf, de orice tip.**

- $\hat{h}_1$(nod)=$\hat{h}$(nod)/3+5
- $\hat{h}_1$(nod)=$\hat{h}$(nod)+0.5
- $\hat{h}_1$(nod)=$\hat{h}$(nod)*1000
- $\hat{h}_1$(nod)=2*$\hat{h}$(nod)^3 (cu sensul de ridicare la puterea a 3-a)
- niciuna dintre formulele de la celelalte subpuncte

4. Care dintre următoarele afirmații sunt adevărate pentru rețele Bayesiene?

- Întotdeauna gradul exterior al oricărui nod (numărul de arce care ies din nod) este mai mare sau egal cu gradul său interior (numărul de arce care intră în nod).
- Întotdeauna gradul interior (numărul de arce care intră în nod) al unui nod este mai mic sau egal cu 1.
- Orice graf orientat cu arce având asociate ponderi poate reprezenta topologia unei rețele Bayesiene.
- Topologia unei rețele Bayesiene este un graf orientat (direcționat).
- Într-o rețea Bayesiană nu pot exista două noduri n_1 și n_2 cu proprietatea că există un drum de la n_1 la n_2 și în același timp există un drum de la n_2 la n_1.

5. Considerăm arborele Minimax de mai jos pentru care cunoaștem valorile din frunze:



Care dintre următoarele afirmații sunt adevărate?

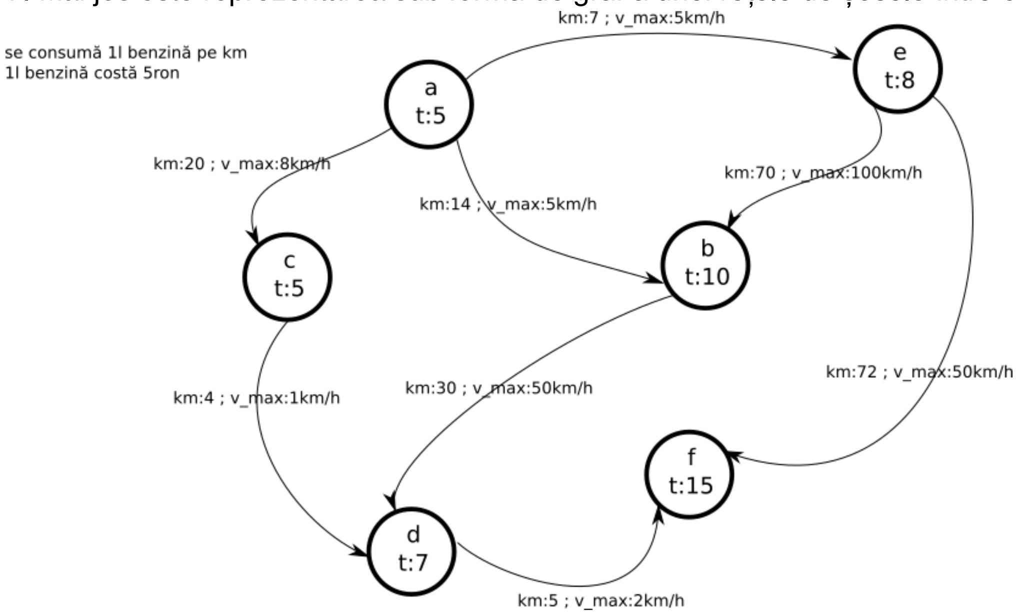
- În nodul ***a*** vom avea valoarea minimax 99.
- Aplicând algoritmul minimax, nodul *h* va avea în mod sigur valoarea 2.
- Dacă valorile frunzelor ar fi toate de 10 ori mai mari (pentru orice nod frunză, în loc de valoarea *v* am avea valoarea 10**v*) atunci setul de noduri din variația principală nu s-ar schimba.

- Dacă am aplica algoritmul alpha-beta pe acest arbore, nodurile D și E nu ar mai fi evaluate.
- Dacă am aplica algoritmul alpha-beta pe acest arbore și valoarea nodului *G* ar fi 55 în loc de 5, atunci nodul *o* nu ar mai fi evaluat.

6. Pentru problema blocurilor (vezi anexa), considerăm o stare cu N stive, dintre care NB sunt stive nevide, și în configurație există un total de B blocuri. Care este numărul de succesori direcți ai acesteia (stări în care se poate ajunge din starea curentă printr-o singură mutare)?

- (N - 1) * B
- N
- N * NB
- (N - 1) * (NB - 1)
- (N - 1) * NB
- N * (NB - 1)

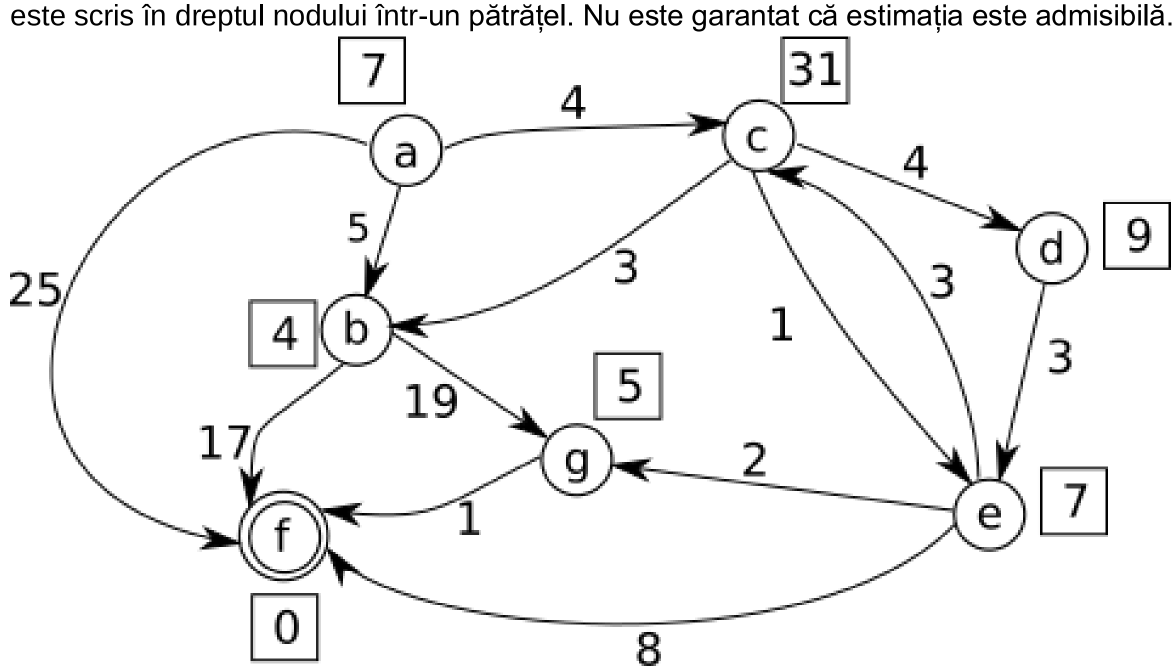
7. Mai jos este reprezentarea sub formă de graf a unei rețele de șosele între orașe.



Orașele sunt reprezentate ca noduri, iar muchiile reprezintă șoselele. Pe fiecare muchie este trecută lungimea sa în kilometri, dar și viteza maximă admisă. La intrarea într-un oraș se plătește o taxă (specificată în fiecare nod prin t:taxă în RON). Dorim să folosim în mod corect A*, cu un cost relevant și o estimație admisibilă, pentru a găsi cel mai bun drum conform unui criteriu specificat mai jos. Nodul ***a*** este nodul de start, și ***f*** este nodul scop. Dorim să găsim cea mai potrivită funcție de cost care asociază unei tranziții n_i->n_j (de la nodul n_i la nodul n_j) un anumit cost. Care dintre următoare afirmații sunt corecte?

- Dacă dorim să găsim drumul cu număr minim de orașe prin care se trece, atunci o funcție potrivită de cost asociază tranziției n_i->n_j numărul nr_km(i,j), unde nr_km(i,j) este numărul de kilometri dintre n_i și n_j.
- Dacă dorim să găsim drumul de timp minim, atunci o funcție potrivită de cost asociază tranziției n_i->n_j numărul de kilometri dintre n_i și n_j.
- Dacă dorim să găsim drumul de cost minim în bani, atunci o funcție potrivită de cost asociază tranziției n_i->n_j numărul de kilometri dintre n_i și n_j.
- Dacă dorim să găsim drumul de timp minim, atunci o funcție potrivită de cost asociază tranziției n_i->n_j valoarea nr_km(i,j)/viteza(i,j), unde nr_km(i,j) e numărul de kilometri dintre n_i și n_j, iar *viteza(i,j)* este viteza maximă admisă pe acel drum.
- Dacă dorim să găsim drumul de cost minim în bani, atunci o funcție potrivită de cost asociază tranziției n_i->n_j valoarea nr_km(i,j)*cost_benzină +taxa(n_j), unde nr_km(i,j) este numărul de kilometri dintre n_i și n_j, cost_benzină este costul unui litru de benzină, iar taxa(n_j) este taxa de intrare în orașul n_j.
- Dacă dorim să găsim drumul de timp minim, atunci o funcție potrivită de cost asociază tranziției n_i->n_j valoarea nr_km(i,j)*viteza(i,j), unde nr_km(i,j) este numărul de kilometri dintre n_i și n_j, iar *viteza(i,j)* este viteza maximă admisă pe acel drum.

8. Se consideră următorul graf cu muchii ponderate (costul unei muchii este scris în dreptul ei). Nodul de start este nodul *a* și nodul scop este *f*. Factorul euristic al fiecărui nod (estimația ĥ) este scris în dreptul nodului într-un pătrățel. Nu este garantat că estimația este admisibilă.



Drumul returnat de A* (și costul lui) folosind estimația din desen este:

- a -> c -> e -> f cu costul 13
- a -> c -> e -> f cu costul 20
- a -> b -> g -> f cu costul 25
- a -> f cu costul 25
- a -> f cu costul 32
- a -> c -> e -> g -> f cu costul 8
- a -> b -> f cu costul 22

9. Considerăm problema 8-puzzle cu următoarea modificare: cele 8 plăcuțe sunt numerotate cu numere de la 1 la 4 și există exact câte două plăcuțe numerotate cu același număr. Un exemplu de stare inițială este:

1	2	1
4	3	4
2	3	

Costul unei mutări de plăcuță este 1.
Scopul este să ajungem la o configurație în care cele două plăcuțe din fiecare pereche (având același număr) sunt vecine pe linie sau coloană. Care dintre următoarele estimații sunt **admisibile**?

- M-1, unde M este minimul dintre distanțele d(i), unde d(i) este distanța Manhattan dintre două plăcuțe inscripționate cu numărul i.
- N/3, unde N este numărul de plăcuțe care nu au niciun vecin cu același număr pe linie sau coloană.
- Numărul de perechi de plăcuțe cu același număr care nu sunt alăturate.
- Numărul de plăcuțe care nu au niciun vecin cu același număr pe linie sau coloană.
- S-1, unde S este suma distanțelor d(i), unde d(i) este distanța Euclidiană dintre două plăcute cu numărul i, considerând drept coordonate pentru distanță linia și coloana fiecărei plăcuțe.
- S-1, unde S este suma distanțelor Manhattan dintre fiecare două plăcuțe inscripționate cu același număr.

10. Considerăm următoarea problemă asemănătoare cu problema blocurilor. Avem un număr N de stive pe care avem așezate blocuri ce conțin **numere naturale nenule**. Se consideră o stare finală (scop) orice stare în care suma numerelor de pe fiecare stivă nevidă este **mai mică sau egală cu un număr *k*** dat în fișierul de intrare. Costul mutării unui bloc este egal cu numărul scris pe el. Vom presupune că niciun bloc nu are un număr mai mare decât *k* înscris pe el.

Care dintre urmatoarele moduri de a calcula estimația ĥ pentru o stare data conduc la o estimație ĥ admisibila?

- Pentru o stivă i, pornind de la baza stivei în sus calculăm suma sb(i) a numerelor înscrise în blocuri până când suma sb(i) ajunge ori să cuprindă toate blocurile de pe stivă ori să depășească numărul k. Dacă suma sb(i) nu cuprinde toate blocurile, atunci pentru fiecare stiva i adunăm la valoarea estimației (inițializată cu 0) suma_totala(i)-sb(i), unde suma_totala(i) reprezintă suma numerelor scrise pe totalitatea blocurilor de pe stiva i.
- Pentru o stivă i, pornind de la vârful stivei în jos calculăm suma sv(i) a numerelor înscrise pe blocuri până când suma sv(i) ajunge ori să cuprindă toate blocurile de pe stivă ori să depășească numărul k. Pentru fiecare stivă i pentru care suma sv(i) nu cuprinde toate blocurile, adunăm la valoarea estimației (inițializată cu 0) suma_totala(i)-sv(i), unde suma_totala(i) reprezintă suma numerelor scrise pe totalitatea blocurilor de pe stiva i.
- Pentru fiecare stivă i calculăm media numerelor înscrise pe blocurile de pe stiva i. Dacă suma de pe stiva i depășește k, atunci pentru toate blocurile cu număr mai mare decât media, adunăm numărul blocului la valoarea estimației (inițializată cu 0).
- Pentru fiecare stivă a cărei sumă depășește k, adunăm la valoarea estimației stării (inițializată cu 0) numărul înscris pe blocul din vârful stivei.
- Pentru fiecare stivă a cărei sumă depășește k, adunăm la valoarea estimației stării (inițializată cu 0) suma numerelor de pe toate blocurile aflate deasupra blocului cu cel mai mare număr de pe acea stivă
- Pentru fiecare stivă a cărei sumă depășește k, adunăm la valoarea estimației stării (inițializată cu 0) diferența S(i)-k, unde S(i) este suma tuturor numerelor înscrise pe blocurile de pe acea stivă

11. Pentru problema X și 0, care dintre următoarele afirmații sunt adevărate?

- O funcție de evaluare minimax relevantă pentru jocul x și 0, pentru noduri nefinale, este NLD(MAX)-NLD(MIN) unde NLD=numarul de linii deschise pentru jucătorul respectiv. O linie deschisă pentru un jucător este o linie care nu conține simbolul jucătorului opus. Prin linie înțelegem oricare dintre rând, coloană sau diagonală în tabla de joc.
- Numărul de succesori (fii direcți) ai unei stări nefinale în arborele minimax este egal cu n, unde n este numărul de locuri libere de pe tabla de joc.
- Presupunem că MAX (calculatorul) folosește simbolul X și utilizatorul simbolul 0. Vom considera următorul mod de calcul al valorii minimax: pentru o stare în care a câștigat MAX, valoarea este +99, pentru o stare în care a câștigat MIN, valoarea e -99, iar pentru remiză e 0. Pentru starea curentă de mai jos, și fără adâncime maximă setată, valoarea minimax va fi +99.

X		
0	X	0
X		0

- Presupunem că MAX (calculatorul) folosește simbolul X și utilizatorul simbolul 0. Vom considera următorul mod de calcul al valorii minimax: pentru o stare în care a câștigat MAX, valoarea este +99, pentru o stare în care a câștigat MIN, valoarea e -99, iar pentru remiza e 0. Pentru starea curentă de mai jos, și fără adâncime maximă setată, valoarea minimax va fi -99.

		X
0	0	
X	X	0

- Considerând simbolul calculatorului ca fiind 0 (deci la începutul jocului utilizatorul mută primul) și numerotarea nivelurilor începând de la 0 (0 fiind nivelul rădăcinii), atunci pentru arborele minimax de adâncime maximă 4 având ca rădăcină tabla:

	X	
X	**0**	

putem găsi o stare finală a jocului (câștig, pierdere sau remiză) pe nivelul 2 al arborelui (se presupune că se aplică algoritmul minimax).