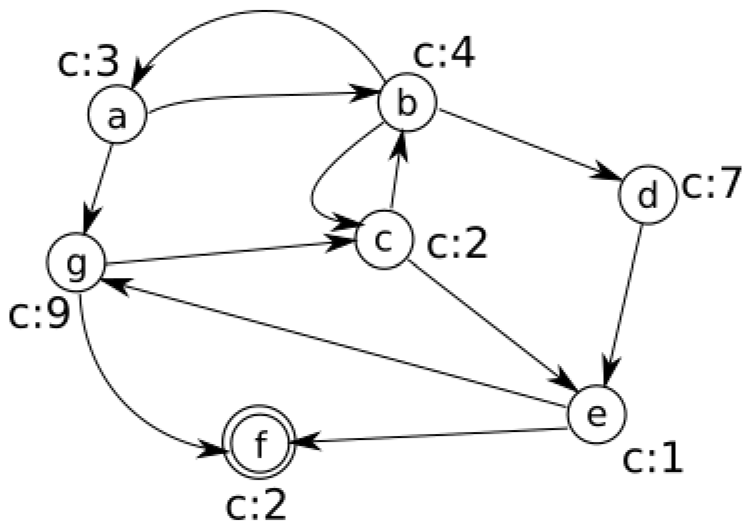
Se consideră graful de mai jos:



Pentru fiecare nod e asociat un cost de intrare în acel nod (prin intermediul notației c: cost). Practic orice tranziție reprezentată printr-un arc care intră în nodul *n* are drept cost asociat costul lui *n*. Costul unui drum e dat de suma costurilor tuturor nodurilor din drum. Care dintre afirmațiile de mai jos sunt adevărate pentru graful dat în imagine:

a. estimația $\hat{h}$ este admisibilă, având valorile: $\hat{h}(a)=3$, $\hat{h}(b)=5$, $\hat{h}(c)=4$, $\hat{h}(d)=1$, $\hat{h}(e)=2$, $\hat{h}(f)=0$, $\hat{h}(g)=3$
b. estimația $\hat{h}$ este admisibilă, având valorile: $\hat{h}(a)=3$, $\hat{h}(b)=3$, $\hat{h}(c)=1$, $\hat{h}(d)=1$, $\hat{h}(e)=2$, $\hat{h}(f)=0$, $\hat{h}(g)=2$
c. estimația $\hat{h}$ este admisibilă, având valorile: $\hat{h}(n)=\{1$ dacă n nu este nod scop, 0 dacă este scop$\}$, unde n e nod din graf
d. estimația $\hat{h}$ este admisibilă, având valoarea: $\hat{h}(n)$=min_cost, iar min_cost=min ({c, unde c e costul unui nod n din graf, oricare ar fi n}) (practic, cel mai mic cost de nod din graf)

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 $\hat{h}$(nod), oricare ar fi aceasta (adică formula trebuie să fie adevărată **pentru orice $\hat{h}$** nu pentru cazuri particulare de $\hat{h}$), o nouă estimație $\hat{h}_1$(nod) **în mod cert admisibilă?**
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.

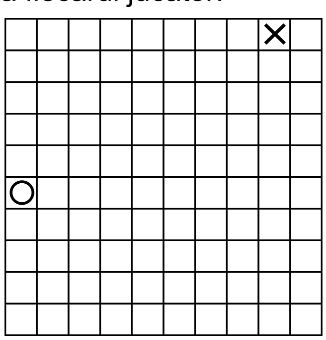
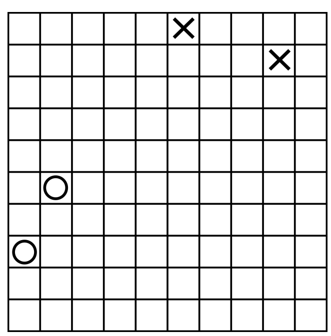
a. $\hat{h}_1(nod)=\hat{h}(nod)/2$
b. $\hat{h}_1(nod)=\hat{h}(nod)*2$
c. $\hat{h}_1(nod)=\hat{h}(nod)+2$, dacă *nod* nu e nod scop și $\hat{h}_1(nod)=0$ dacă *nod* este scop
d. $\hat{h}_1(nod)=\max(0, \hat{h}(nod)-3)$
e. niciuna dintre formulele de la celelalte subpuncte

Se consideră următorul joc. Avem un grid cu N linii și N coloane, inițial vid.Sunt doi jucători, primul jucător folosește x și al doilea simbolul 0.

Regulile jocului sunt următoarele:

- jucătorul cu simbolul x mută primul
- O mutare e formată din două acțiuni. Plasarea unui simbol propriu și deplasarea tuturor simbolurilor cu o poziție pe direcția de deplasare a jucătorului curent.
- jucatorul x poate plasa simboluri doar pe prima linie, în locuri libere, cu excepția ultimei poziții din linie
- jucătorul 0 poate plasa simboluri doar pe prima coloană cu excepția ultimei poziții din coloană
- jucătorii își pot muta simbolurile proprii cu o singura pozitie doar în direcția opusă configurației inițiale (x poate muta doar în jos pe coloană iar 0 poate muta doar în dreapta pe linie). Jucătorii sunt obligați să facă o mutare. Dacă un jucător nu poate muta când îi vine rândul, atunci e remiză.
- dacă în deplasarea unui simbol, acesta ar trebui să se mute în locul unei piese adversare, piesa adversară e capturată.
- când un simbol ajunge în capătul opus celui din care a pornit, se oprește.
- dacă în deplasarea unui simbol, acesta ar trebui să se mute în locul unei piese prietene (a aceluiași jucător), simbolul nu se mai mută (rămâne blocat)
- în momentul în care un jucător a umplut linia/coloana opusă linie/coloanei din care îi pornesc simbolurile, atunci a câștigat.
- Totuși, dacă un jucător cucerește M simboluri, atunci pierde (unde M e o constantă a jocului).

În imaginile de mai jos se vede prima mutare atât a lui X cât și a lui 0. Respectiv a doua mutare a fiecărui jucător:

Care dintre următoarele fraze sunt adevărate:

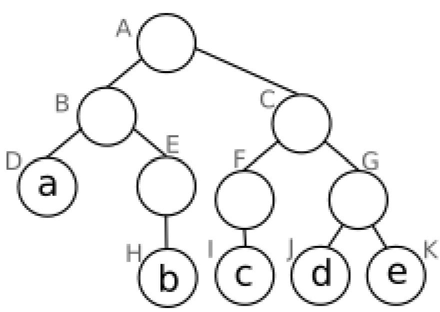
a. Orice nod itern(care nu e frunză) în arborele minimax are N succesori
b. O funcție de evaluare minimax relevantă pentru jocul prezentat este NMAX-NMIN, unde NMAX reprezintă numărul de simboluri ale lui MAX pe linia/coloana opusă liniei/coloanei de început, iar NMIN reprezintă numărul de simboluri ale lui MIN pe linia/coloana opusă liniei/coloanei de început

c. O funcție de evaluare minimax relevantă pentru jocul prezentat este NPIESEMAX-NPIESEMIN, unde NPIESEMAX reprezintă numărul de simboluri ale lui MAX pe tablă și NPIESEMIN numărul de simboluri ale lui MIN pe tablă
d. O funcție de evaluare minimax relevantă pentru jocul prezentat este NCUCERITEMIN-NCUCERITEMAX, unde NCUCERITEMAXreprezintă numărul de simboluri cucerite ale lui MAX pe tablă și NCUCERITEMIN numărul de simboluri cucerite ale lui MIN pe tablă

Care dintre următoarele fraze sunt adevărate:
a. În cadrul reprezentării cunoștințelor, moștenirea proprietăților nu este o formă de inferență
b. Nu este necesar pentru sistemul expert să știe să explice răspunsurile date
c. "slot filler" este o tehnică de reprezentare a cunoștintelor
d. Orice nod al unei rețele Bayesiene are proprietatea că este o variabilă aleatoare.

Pentru un arbore minimax de adâncime maximă A generat pentru un joc oarecare, care dintre următoarele afirmații sunt adevărate? Se consideră că rădăcina e la adâncime 0, iar nodurile de pe cel mai de jos nivel au adâncime A.
a. Fiii unui nod MAX (dacă există) sunt întotdeauna noduri MIN
b. Dacă A e impar toate frunzele arborelui sunt MIN
c. Dacă A e par frunzele de pe cel mai de jos nivel sunt noduri MAX
d. Rădăcina arborelui MINIMAX întotdeauna va avea fii.
e. Dacă un jucător are întotdeauna maxim N mutări, numărul de noduri din arbore nu va depăși N^A

Se consideră arborele minimax din imagine. Numele nodurilor sunt scrise cu gri lângă cercul fiecărui nod. Valorile minimax ale nodurilor-frunză sunt trecute în interiorul cerculețelor și sunt simbolizate prin litere mici ale alfabetului.



Care dintre următoarele afirmații sunt adevărate:
a. Aplicând algoritmul Alpha-Beta, nodul D nu va fi niciodată eliminat (retezat)

b. **Aplicând algoritmul Alpha-Beta, dacă $a>c$ și $b>c$ atunci nodul G împreună cu subarborele său va fi retezat**
c. Aplicând algoritmul Alpha-Beta, dacă $a>c$ și $b>c$ atunci nodul F împreună cu subarborele său va fi retezat
d. **Aplicând algoritmul Alpha-Beta, dacă $c<d$ atunci nodul K va fi eliminat (retezat)**

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 tote blocurile de pe o stivă au doar numere cu aceeași paritate. Costul mutării unui bloc este egal cu nivelul la care se află pe stivă (de exemplu dacă stiva are înălțime k - adică sunt k blocuri pe stivă, atunci costul mutării blocului din vârf e k; considerăm blocul de la bază aflat la nivelul 1).

Care dintre urmatoarele moduri de a calcula estimația $\hat{h}$ pentru o stare data conduc la o estimație $\hat{h}$ admisibila?

a. Pentru fiecare stivă i, adunăm la valoarea estimației (inițializată cu 0) înălțimea stivei i
b. **Pentru fiecare stivă i, cu numere de parități diferite, adunăm la valoarea estimației (inițializată cu 0) numărul 2.**
c. **Pentru fiecare stivă i, cu numere de parități diferite, adunăm la valoarea estimației (inițializată cu 0) numărul ni unde ni este numărul de blocuri impare de pe stivă.**
d. **Pentru fiecare stivă i, cu numere de parități diferite, adunăm la valoarea estimației (inițializată cu 0) numărul ni unde ni este numărul de blocuri pare de pe stivă.**
e. Pentru fiecare stivă i, cu numere de parități diferite, adunăm la valoarea estimației (inițializată cu 0) numărul ni unde ni este suma nivelelor blocurilor impare de pe stivă.

Considerăm problema 8-puzzle cu următoarea modificare: cele 8 plăcuțe sunt notate cu numere naturale nenule oarecare (nu neapărat de la 1 la 8) iar mutarea se face fie mutând o plăcuță într-un loc liber, fie sărind cu o placuță peste altă plăcuță în locul liber, iar placuta peste care se sare dispare. Un exemplu de stare inițială este:

1	7	1
12	10	2
5	3	

Costul unei glisări de plăcuță este 1, iar al unei sărituri este egal cu numărul inscripționat pe plăcuța care sare.
Scopul este să ajungem la o configurație în care mai avem doar o plăcuță. Care dintre următoarele estimații sunt **admisibile indiferent de ce numere avem înscrise pe plăcuțe**?

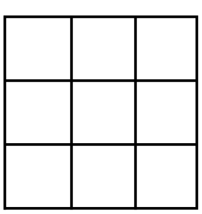
- **a. N-1, unde N e numărul de plăcuțe de pe tablă.**
- b. Numărul maxim care se găsește pe tablă.
- c. S-NMIN, unde S este suma numerelor de pe plăcuțe iar NMIN este numărul minim din tablă.
- d. S-NMAX, unde S este suma numerelor de pe plăcuțe iar NMAX este numărul maxim din tablă.
- **e. (N-1)*NMIN unde NMIN este numărul minim de pe tablă, iar N e numărul de plăcuțe de pe tablă**

Care dintre următoarele afirmații sunt adevărate cu privire la algoritmul A*:

- **a. Nodul de start va fi expandat o singură dată în cadrul algoritmului.**
- b. Dacă o stare a ajuns în lista closed, atunci măsurile $\hat{f}$ și g pentru informația acelei stări rămân fixe (nu mai pot fi actualizate pentru acea informație)
- c. Considerăm nodurile n1 și n2 cu măsurile g1,$\hat{h}$1,$\hat{f}$1, respectiv, g2, $\hat{h}$2, $\hat{f}$2 (notațiile $\hat{f}$,g,$\hat{h}$ sunt cele conform algorimtului A*). Dacă g1<g2 atunci obligatoriu nodul n1 se află înaintea nodului n2 în open.
- d. Indiferent de tipul de estimație (euristică), nu există niciun nod în open care să aibă $\hat{f}$-ul mai mic decât $\hat{f}$-ul oricărui nod din closed.
- e. Imediat ce nodul scop a intrat în coada OPEN, putem returna drumul soluție de cost minim.

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

- a. Pentru starea inițială a tablei de joc, reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 9! (9 factorial = 1*2*3*4*5*6*7*8*9).



b. Pentru starea inițială a tablei de joc, reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 1+(9+8*9+7*8*9+6*7*8*9+....+1*2*3*4*5*6*7*8*9)

c. 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.

d. O funcție de evaluare minimax relevantă pentru jocul x și 0 este NLD(MAX)-NLD(MIN) cand e rândul pentru MAX să mute, respectiv NLD(MIN)-NLD(MAX) când e rândul lui MIN să mute. Am notat cu NLD numărul 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.

e. Orice stare finală a jocului are proprietatea că tabla este plină (nu mai are spații libere).

f. 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.

g. 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

h. 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	X	0
X		0

i. **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

j. 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

k. Presupunem că MAX (calculatorul) foloseste 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 0.

		X
0	0	
X	X	0

I. 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).