1. Se consideră problema blocurilor (cu S stive și NB blocuri; S și NB naturale, S>=3, NB>=4, NB par) cu următoarele modificări:

- NB-1 blocuri conțin o informație numerică de forma 0 sau 1.
- Există un bloc special, numit blocul "not", peste care nu poate exista niciodată un alt element și care nu se poate găsi niciodată pe stivele din capetele configurației
- Ca la problema blocurilor clasică, putem **muta** un bloc dintr-un vârf de stivă pe alt vârf de stivă, cu următoarele restricții:
  - Blocul "not" se poate muta pe orice stivă mai puțin cele din capete (prima și ultima).
  - Când blocul "not" este mutat pe o stivă nevidă, valoarea tuturor blocurilor de pe acea stivă se modifică din 0 în 1 și din 1 în 0.
  - Peste blocul "not" nu putem muta niciun bloc numeric.
- **Costul** mutării unui *bloc numeric* este egal cu numărul de pe bloc + 1 (blocurile 0 au costul 1, blocurile 1 au costul 2). Costul mutării blocului "not" este 1 + nr_schimbari, unde nr_schimbari e numărul de blocuri numerice afectate de mutarea sa.
- **Scopul** este să ajungem la o configurație în care există exact o stivă doar cu blocuri 0, exact o stivă doar cu blocuri 1, iar ambele stive conțin exact NB/2 blocuri (NB precizat la început).

Notăm cu SV numărul de stive vide.

Care dintre următoarele afirmații sunt adevărate?

a. **Pentru S>=4, nu există nicio stare inițială pentru care să nu existe soluție.**
b. **Pentru S=3, există măcar o stare inițială în mulțimea de stări inițiale posibile (cu parametrii conform condițiilor din enunț) pentru care să nu putem găsi soluție folosind A*.**
c. Numărul de succesori valizi ai unei stări oarecare este 2*(S-SV-2)
d. **Presupunând că blocul "not" se află singur pe o stivă (sub el nu mai e alt bloc). Numărul de succesori valizi ai unei stări oarecare este (S-SV-1)*(S-2)+S-3**
e. Presupunând că blocul "not" se află singur pe o stivă (sub el nu mai e alt bloc). Numărul de succesori valizi ai unei stări oarecare este (S-SV-2)*(S-1)+S-3
f. Presupunând că blocul "not" se află singur pe o stivă (sub el nu mai e alt bloc). Numărul de succesori valizi ai unei stări oarecare este (S-1)+(S-SV)*(S-1)
g. **Presupunând că blocul "not" se află singur pe o stivă (sub el nu mai e alt bloc). Numărul de succesori valizi ai unei stări oarecare este (S-SV)*(S-2)-1**
h. Presupunând că blocul "not" se află singur pe o stivă (sub el nu mai e alt bloc). Numărul de succesori valizi ai unei stări oarecare este (S-SV)*(S-2)+1
i. Pentru S=3 nu putem avea o stare inițială conținând în configurație o stivă vidă.
j. niciuna dintre opțiunile de mai sus nu e corectă.

Using Custom Setup

2. Considerăm enunțul problemei anterioare. Care este o estimație admisibilă pentru o stare?

a. **S-SV-3, unde SV e numărul de stive vide din starea respectivă.**
b. SV, unde SV e numărul de stive vide din starea respectivă.
c. |Nr(0)-Nr(1)|, unde Nr(x) e numărul de blocuri cu valoarea x.
d. **|Nr(0)-Nr(1)|/2, unde Nr(x) e numărul de blocuri cu valoarea x.**
e. **0 indiferent de stare.**
f. **1 dacă starea nu e scop și 0 dacă e stare scop.**
g. 1 indiferent de stare.
h. niciuna dintre opțiunile de mai sus nu e corectă.

3. Considerăm un arbore minimax de adâncime maximă A și N noduri (inclusiv rădăcina) generat pentru un joc oarecare. Care dintre următoarele afirmații sunt adevărate?

a. **Un nod MIN nu poate fi frate cu un nod MAX**
b. Dacă aplicăm alpha-beta pe aceeași stare rădăcină putem să evităm calcularea a maxim N/2 noduri
c. **Dacă aplicăm alpha-beta pe aceeași stare rădăcină putem să evităm calcularea a maxim N-1 noduri**
d. Dacă aplicăm alpha-beta pe aceeași stare rădăcină putem întotdeauna să evităm calcularea a minim unui nod
e. **Alpha-beta nu va elimina (reteza) niciodată primul fiu al unui nod.**
f. Frunzele sunt întotdeauna noduri MIN.
g. Frunzele sunt întotdeauna noduri MAX.
h. niciuna dintre opțiunile de mai sus nu e corectă.

5. Pentru *N* număr natural (5 ≤ *N* ≤ 10) considerăm o tablă de joc de dimensiune NxN în care sunt *NL* lăzi notate pe hartă cu câte un număr *G* reprezentând greutatea fiecărei lăzi. *NL* și *G* sunt numere naturale nenule. Pe hartă avem și un omuleț notat cu *X*, așa cum se vede în desen (desenul arată doar un exemplu de stare inițială; exercițiul nu se referă doar la acest caz specific de stare inițială).

	3			10
				8
	5			
X	14			

Scopul omulețului este să împingă un număr K de lăzi (K natural nenul, K <= NL) în colțul din stânga sus al matricei (poziția 0,0). Omulețul poate împinge o ladă doar cu câte o poziție și doar într-un spațiu liber. În plus, el trebuie să fie în spațiul vecin lăzii pe aceeași direcție dar în sens opus față de sensul mutării (de exemplu, pentru a muta o ladă în dreapta, omulețul trebuie să fie în stânga ei; cum e pe desen în cazul lăzii 14). În starea inițială pe poziția (0,0) nu poate exista nicio ladă și aceasta nu poate reprezenta nici poziția inițială a omulețului. O ladă ajunsă (împinsă) în locația (0,0) dispare. Omulețul se poate deplasa și poate muta lăzi doar cu câte o poziție în direcțiile: sus, jos, stânga și dreapta (nu și pe diagonală). Fiecare deplasare **costă**:

- 1 dacă omulețul nu împinge o ladă
- greutatea lăzii dacă omulețul împinge o ladă

Notații:

- Coordonatele omulețului: linie(om), coloana(om).
- Coordonatele unei lăzi: linie(lada[i]) și coloana(lada[i]).
- Distanța Manhattan: distMH(linie1,coloana1,line2,coloana2)=|linie1-linie2|+|coloana1-coloana2|
- Greutatea lăzii x: greutate(x)

Care dintre estimațiile de mai jos sunt admisibile pentru această problemă?

a. **Dacă omulețul a scos deja K1 lăzi din hartă în starea curentă, o estimație admisibilă este K - K1**
b. O estimație admisibila este distMH(linie(om), coloana(om), 0, 0) indiferent de câte lăzi au fost scoase de pe hartă
c. **Notând cu min_lazi={greutatea minimă dintre greutățile lăzilor aflate pe hartă), și cu dist_min={cea mai mică distanță Manhattan la care se afla o ladă față de colțul stânga-sus}, o estimație admisibilă este dist_min*min_lazi dacă omulețul mai are de scos lăzi din configurație și 0 dacă nu.**
d. Notând cu min_lazi={greutatea minimă dintre greutățile lăzilor aflate pe hartă), și cu dist_min={cea mai mica distanta Manhattan la care se afla o lada de colțul stânga-sus}, o estimație admisibilă este dist_min*min_lazi*K
e. **Pentru NL>=4 există o configurație inițială (în spațiu stărilor posibile) fără soluție.**
f. Pentru NL=1 problema întotdeauna are soluție.
g. O estimație admisibila pentru o stare în care omul mai are de scos un număr nenul de lăzi, este distMH(linie(om), coloana(om), linie(lada_min), coloana(lada_min))*greutate(lada_min) , unde lada_min e lada cu cea mai mică greutate din configurație
h. niciuna dintre opțiunile de mai sus nu e corectă.

5. Care dintre următoarele afirmații sunt adevărate pentru problema X și 0? (enunț în anexă)

a. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 6=3!

x	0	
	x	
x	0	

b. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 24=4!

x	0	
	x	
x	0	

c. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, adâncimea arborelui este 1

x	0	
	x	
x	0	

d. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, adâncimea arborelui este 1

0	**x**	**x**
	0	**x**
0	**x**	

e. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 28

x	**0**	**x**
	x	
	0	

f. O stare finală a jocului are ca proprietate că fie a câștigat X, fie a câștigat 0.

g. Arborele minimax pentru X și 0 întotdeauna va avea cel mult 10 niveluri.

h. Pentru starea reprezentată mai jos, calculând arborele minimax în totalitate, fără a impune o adâncime maximă, numărul de noduri din arbore ar fi egal cu 5

x	0	x
0	x	
	0	x

i. Un nod frunză din arborele Minimax, pentru jocul x și 0 conține întotdeauna o tablă fără locuri libere (tabla e în întregime completată).

j. niciuna dintre opțiunile de mai sus nu e corectă.

6. Se dă o problemă de căutare cu costuri strict pozitive pe tranziții. Se dau o estimație *admisibilă* $\hat{h}1$(nod) și o estimație *neadmisibilă* $\hat{h}2$(nod), oarecare. Care dintre formulele următoare ar obține din $\hat{h}1$ și $\hat{h}2$ o nouă estimație $\hat{h}$(nod) cu siguranță **neadmisibilă**?

Observație: Formula trebuie să fie adevărată pentru orice $\hat{h}1$, orice $\hat{h}2$ și orice nod din graf, nu doar pentru cazuri particulare. Știm că $\hat{h}1(nod) \geq 0$ și $\hat{h}2(nod) \geq 0$ pentru orice nod din graf.

a. **$\hat{h}(nod) = \hat{h}2(nod) + \hat{h}1(nod) / 3$**
b. $\hat{h}(nod) = \hat{h}1(nod) + \hat{h}2(nod) / 3$
c. $\hat{h}(nod) = \hat{h}2(nod) - \hat{h}1(nod)$

d. **$\hat{h}(nod) = 2*\hat{h}2(nod) - \hat{h}1(nod)$**
e. **$\hat{h}(nod) = \max(\hat{h}2(nod), \hat{h}1(nod))$**
f. $\hat{h}(nod) = \min(\hat{h}2(nod), \hat{h}1(nod))$
g. $\hat{h}(nod) = (\hat{h}1(nod) + \hat{h}2(nod)) / 2$
h. $\hat{h}(nod) = \hat{h}1(nod) * \hat{h}2(nod)$
i. niciuna dintre formulele de mai sus nu e corectă

7. Considerăm urmatorul joc:
Pentru N număr natural ($5 \leq N \leq 10$) considerăm o tablă de joc de dimensiune NxN și 2 jucători (X și 0). Jucătorului X îi corespunde prima linie, iar lui 0 ultima. Jucătorul X mută primul.
O mutare constă în una dintre următoarele acțiuni:

1. Plasarea unui simbol nou pe linia proprie.
2. Mutarea unui simbol cu o singură poziție pe orizontală (în stânga sau dreapta) sau pe verticală, înspre linia adversarului (în jos pentru x și în sus pentru 0).

Restricții de mutare:

1. Nu pot exista mai mult de 2 simboluri ale aceluiași jucător pe vreo linie
2. Plasarea unui simbol nou se poate realiza doar dacă pe coloana respectivă nu mai sunt simboluri proprii.

Jucătorul *j* poate captura o piesă a adversarului, *ja*, dacă piesa adversarului se află între două piese ale jucătorului *j* aflate pe aceeași linie cu aceasta, iar cele două piese ale jucătorului *j* au între ele cel mult N/3 spații.
Un jucător **câștigă** fie când a capturat un număr K_MAX > 3 de simboluri, fie când a ajuns cu un simbol pe linia adversarului.
Care dintre variantele de mai jos oferă o funcție de evaluare care să arate în mod corect cât de favorabilă este starea jocului pentru calculator (MAX) (să aibă o valoare mai mare pentru stări mai favorabile și mai mică pentru stări mai nefavorabile) ?

a. Numărul de simboluri ale lui MAX de pe linii pare din care scădem numărul de simboluri ale lui MIN de pe coloane impare.
b. Câte configurații de 3 simboluri pe poziții consecutive (pe linie/coloană/diagonală) are MAX din care scădem câte configurații de 3 simboluri are MIN
c. **Numărul de simboluri capturate ale lui MAX din care scădem numărul de simboluri capturate ale lui MIN.**
d. **Considerăm LMIN linia celui mai depărtat simbol al lui MIN de linia sa de start și START_MIN, linia de start a lui MIN. Considerăm LMAX linia celui mai depărtat simbol al lui MAX de linia sa de start și START_MAX, linia de start a lui MAX. Atunci funcția de evaluare ar fi | LMAX-START_MAX | - | LMIN-START_MIN |**
e. Numărul de simboluri ale lui MAX la care adunăm numărul de simboluri ale lui MIN.
f. Numărul de simboluri capturate ale lui MAX înmulțite numărul de simboluri capturate ale lui MIN.
g. Numărul de locuri libere de pe tablă.

8. Pentru graful din imagine avem stările cozii OPEN **ordonate** pentru algoritmul A* de mai jos. În fiecare stare sunt precizate doar informațiile nodurilor din coadă. Considerăm nodul **start a**, iar nodul **scop f**. Prima stare e cea inițială a cozii OPEN (cu nodul start inclus). Toate celelalte stări sunt afișate în urma repetării următoarelor acțiuni:

1) Extrage nodul cu f̂ minim (nu se afișează coada pentru această stare intermediară)

2) Adaugă succesorii în coadă astfel încât coada să fie mereu ordonată crescător după f̂ (după această acțiune se afișează coada).

Observatie: Pentru valori f̂ egale, elementele sunt în ordinea descrescătoare a valorii g. În cazul în care și f̂ și g sunt egale, nodurile se ordonează după informație. În cazul în care un succesor are deja informația în CLOSED, dar găsim un cost mai bun pentru aceeași informație, îl mutăm înapoi în OPEN pentru expandare.

Stările cozii OPEN (numărul din stânga e numărul de ordine al stării):

0: [a]

1: [c, b, g]

2: [b, g]

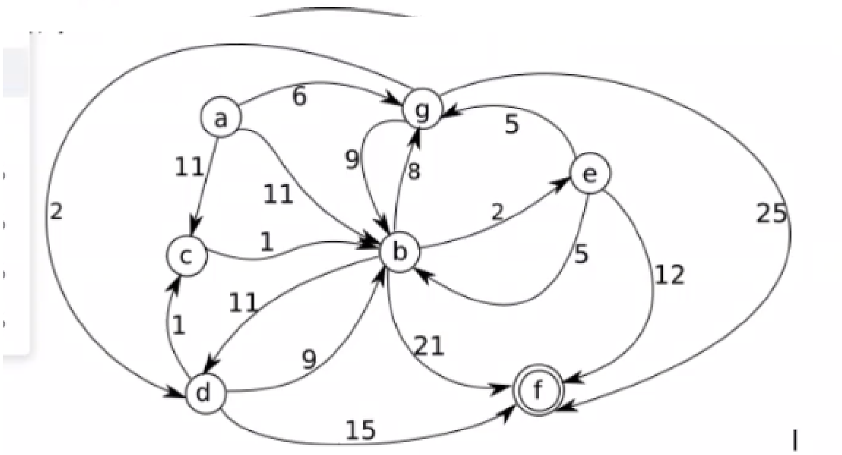
3: [g, e, f, d]

4: [d, e, f]

5: [c, f, e]

6: [b, f, e]

7: [f, e]



Care dintre următoarele afirmații sunt adevărate pentru **graful dat și** **stările cozii OPEN de mai sus**?

a. Estimația (cu valorile scrise între paranteze drepte pentru fiecare nod) a[10] b[1] c[3] d[5] e[4] f[0] g[6]) este admisibilă și în concordanță cu coada OPEN.

b. Estimația (cu valorile scrise între paranteze drepte pentru fiecare nod) a[10] b[10] c[3] d[10] e[11] f[0] g[17]) este admisibilă și în concondanță cu coada OPEN.

c. Estimația (cu valorile scrise între paranteze drepte pentru fiecare nod) a[0] b[0] c[0] d[0] e[0] f[0] g[0]) este admisibilă și în concordanță cu coada OPEN.

d. Estimația (cu valorile scrise între paranteze drepte pentru fiecare nod) a[1] b[1] c[1] d[1] e[1] f[1] g[1] este admisibilă și în concordanță cu coada OPEN.

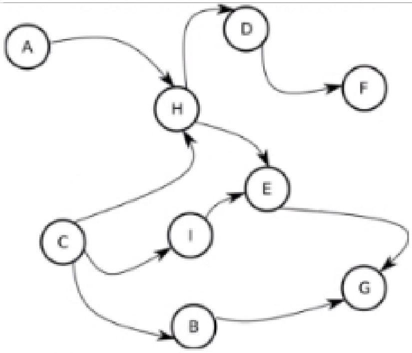
e. Presupunând că orice nod extras din OPEN e imediat adăugat în CLOSED, în starea 3 avem în lista CLOSED (nu neapărat în ordinea asta) nodurile a,b,c

f. În trecerea de la starea 2 la starea 3 a cozii OPEN se expandează nodul b

g. În trecerea de la starea 6 la starea 7 a cozii OPEN se expandează nodul f

h. Niciuna dintre opțiunile de mai sus nu e corectă.

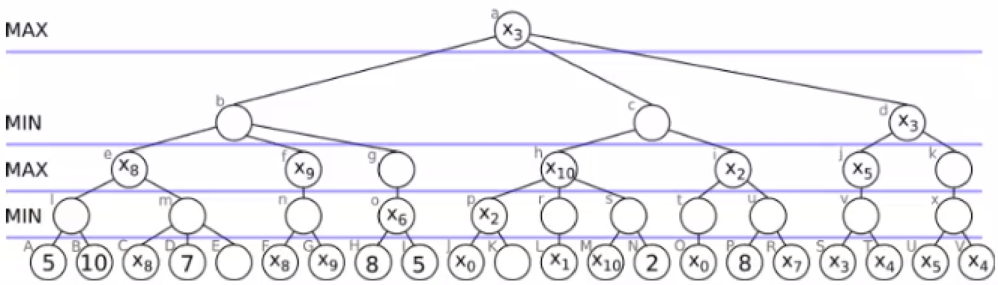
9. Se da urmatoarea topoligie de retea Bayesiana:



Care dintre următoarele afirmații sunt adevărate?

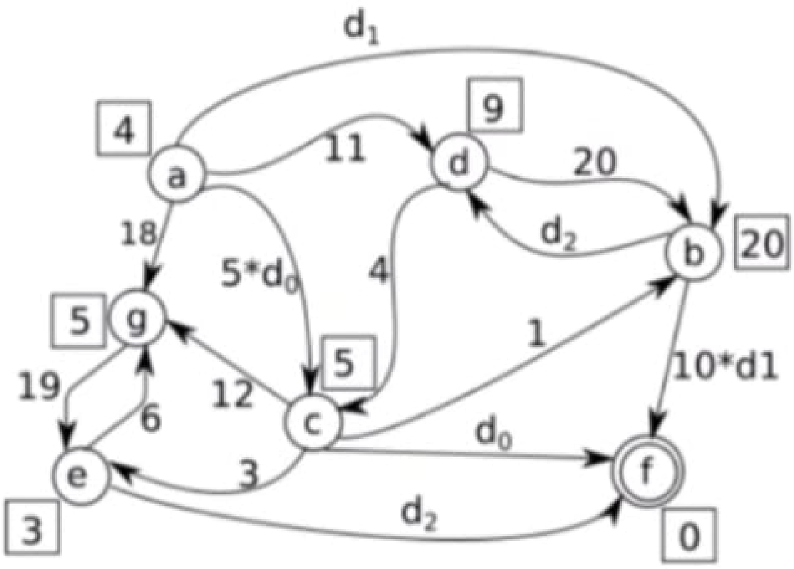
a. Mulțimea {I,B,E} d-separă mulțimea {C} de mulțimea {E,G}.
b. Dacă am adăuga arcul G->C graful nu ar mai fi o topologie corectă pentru o rețea Bayesiana.Mulțimea {I,B,E} d-separă mulțimea {C} de mulțimea {E,G}.
c. Graful dat nu este o topologie corectă pentru o rețea Bayesiană.
d. **Dacă am adăuga arcul C->A graful ar fi în continuare o topologie corectă pentru o rețea Bayesiana.Mulțimea {D} d-separă mulțimea {A} de mulțimea {F}**
e. **Dacă am adăuga arcul G->C graful nu ar mai fi o topologie corectă pentru o rețea Bayesiana.**
f. **Drumul de la nodul A la nodul F e blocat conditionat de mulțimea {D}**
g. Dacă am șterge nodul H, graful nu ar mai fi o topologie corectă pentru o rețea Bayesiana.Mulțimea {I,B,E} d-separă mulțimea {C} de mulțimea {E,G}.
h. **Mulțimea {D} d-separă mulțimea {A} de mulțimea {F}**
i. niciuna dintre opțiunile de mai sus nu e corectă.

10. Se consideră arborele minimax din imagine, cu adâncime maximă 4 (rădăcina fiind considerată la adâncime 0). Se presupune că arborele a fost deja generat prin minimax, iar unele valori minimax au fost, apoi, fie șterse din imagine (nodurile fără conținut), fie înlocuite cu identificatori xi (două noduri cu același identificator xi au valorile minimax egale, însă putem avea xi==xj pentu i diferit de j). Care dintre afirmațiile de mai jos sunt sigur adevărate având în vedere informația dată despre arbore?
Observație: frunzele sunt notate cu litere mari iar nodurile interne cu litere mici.



a. **x2 este sigur egal cu x0**
b. x2 este sigur mai mare sau egal cu 8
c. **x10 e sigur mai mare sau egal cu x1**
d. **nodul c (literă mică) va avea sigur valoarea x2**
e. nodul c (literă mică) va avea sigur valoarea x10
f. **nodul k (literă mică) va avea sigur valoarea x5**
g. **valoarea x5 este întotdeauna egală cu valoarea x3**
h. valoarea x5 este întotdeauna egală cu valoarea x4
i. **x6 sigur are valoarea 5**
j. x7 este sigur mai mic strict decât x2
k. **x8 aparține sigur intervalului [5,7]**
l. **nodul b are sigur valoarea x9**
m. x3 este cea mai mare valoare dintre toate valorile nodurilor arborelui
n. niciuna dintre opțiunile de mai sus nu e corectă.

11. Se dă graful orientat cu arce ponderate din imagine. Pentru unele arce ponderile nu sunt precizate, fiind înlocuite de identificatorii di. Euristica dată este una admisibilă (estimația pentru fiecare nod e trecută în pătrățelul de lângă nod).
Costurile arcelor sunt numere naturale nenule.
Nodul start este a și nodul f este nod scop.
Drumul de cost minim returnat de A* pentru nodul de start a este a->b->d->c->f cu costul 14 și este unicul drum de acest cost.



Care dintre afirmațiile de mai jos sunt adevărate:

a. d0 are valoarea 5
b. d0 poate avea orice valoare mai mare sau egală cu 6.
c. d2 este 3
d. d2 este întotdeauna mai mic strict decât 2
e. d1+d2=5 întotdeauna
f. d1 este 4 și d2 este 1
g. d1 este 1
h. d1 este 2
i. există în graf un drum cu costul 15
j. niciuna dintre variantele de mai sus nu este corectă

12. Care dintre următoarele afirmații sunt adevărate pentru rețele Bayesiene (și topologiile lor)?

a. **O proprietate a topologiei unei rețele Bayesiene este faptul că este un graf orientat (direcționat).**

b. **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.**

c. Întotdeauna gradul interior (numărul de arce care intră în nod) al unui nod este mai mic sau egal cu 1.

d. 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).

e. Orice graf orientat cu arce având asociate ponderi numere reale poate reprezenta topologia unei rețele Bayesiene.

f. **O proprietate a topologiei unei rețele Bayesiene este faptul că este un graf fără circuite (aciclic).**

g. O proprietate a topologiei unei rețele Bayesiene este faptul că este un arbore.

h. **Nodurile unei rețele Bayesiene sunt variabile aleatoare.**

i. Fiecărei conexiuni dintre nodurile unei rețele Bayesiene îi corespunde un tabel de probabilități condiționate care cuantifică efectele pe care părinții le au asupra nodului respectiv

j. **Fiecărui nod al unei rețele Bayesiene îi corespunde un tabel de probabilități condiționate care cuantifică efectele pe care părinții le au asupra nodului respectiv**

k. niciuna dintre opțiunile de mai sus nu e corectă.