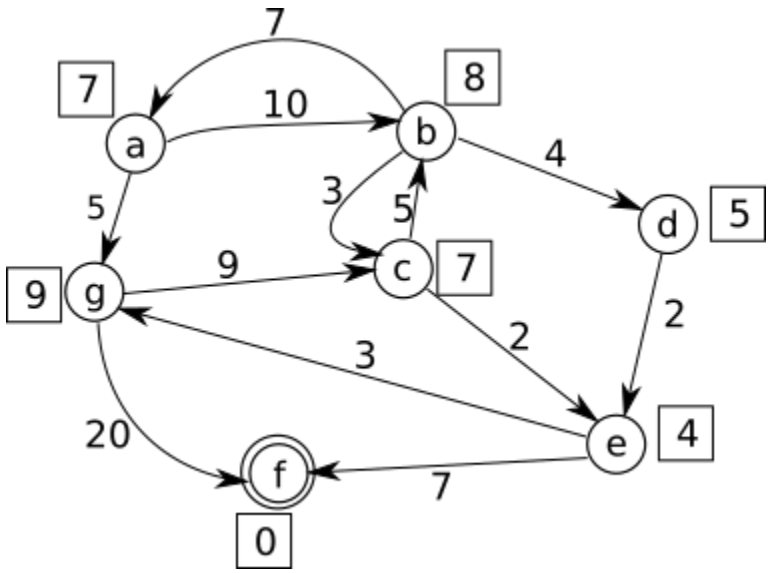
Exemplu de enunț cu rezolvare

Enunțuri:

I) Se dă graful de mai jos, cu următoarele caracteristici:
- Nodul *a* este nodul de start
- Nodul cu cerc dublu este nodul scop
- Numărul înscris lângă fiecare arc este costul acelui arc
- Numărul înscris în pătratul de lângă fiecare nod este h estimat (euristica).



Cerințe (**este bine să faceți subpunctele 1 si 2 împreună pentru a putea verifica dacă un nod nu există deja în drumul la care tocmai vreți să îl adăugați; practic cu ajutorul arborelui vedeți drumurile curent extinse. Atât la examen cât și în temă este bine să faceți arborele pe o foaie separată ca să îl desenați în timp ce scrieți listele open și closed de la fiecare pas**):

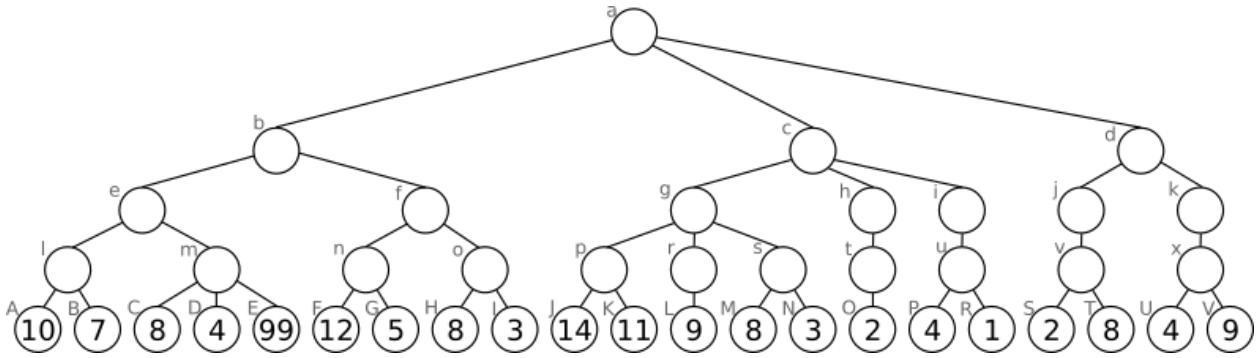
1) Aplicați algoritmul A* pe acest graf precizând următoarele:
- Cum se inițializează listele **open** și **closed**.
- Descrierea fiecărei iterații (cum se modifică listele open și closed). Pentru fiecare nod din listele open și closed se vor scrie următoarele informații în formatul următor (litera nodului, g-ul, f-ul estimat, părintele în arbore). Dacă o informație de nod este readusă din closed în open, trebuie specificat clar acest lucru și explicat de ce se întâmplă asta.
- Scrierea concluziei: care este drumul de cost minim și să se precizeze costul acestuia.

2) Desenați arborele asociat parcurgerii. Pentru fiecare nod scrieți g-ul și f-ul (sub forma unei etichete scrise lângă nodul corespunzător). În arbore se vor reprezenta toate nodurile parcurse și se vor tăia cele care cu o informație pentru care s-a găsit o rută mai bună (au fost înlocuite în coadă de un nod cu aceeași informație dar cu cost mai mic)..

3) Cum ați putea modifica minimal costurile muchiilor (modificați costul a cât mai puține muchii) astfel încât euristica să fie neadmisibilă.

4) Desenați primii 3 arbori generați de IDA*, precizând limita de cost pentru fiecare și arătând modificările prin care trece stiva, evidențiând și întoarcerile.

II) Se dă arborele Minimax (generat de calculator pentru determinarea următoarei mutări) din figură pentru care cunoaștem valorile frunzelor:



1) Etichetați nivelele cu MIN și MAX (atenție la rădăcină!). Completați valorile nodurilor conform algoritmului Minimax. Indicați valoarea jocului și variația principală.
2) Aplicați acestui arbore Algoritmul Alpha-Beta. Desenați arborele rezultat (pe arbore să se vadă cum s-au actualizat la fiecare pas informațiile nodurilor) și explicați operațiile de alpha-beta retezare care au fost efectuate.
3) Reordonați succesorii unui nod din arbore, astfel încât numărul de retezări efectuate de Alpha-Beta să fie mai mare
4) Schimbați valoarea unui singur nod frunză astfel încât să se schimbe valoarea jocului și variația principală să includă alte noduri. Desenați arborele rezultat.

Rezolvări:

I)
1)
Pas 1.
Inițializări:
Open: [(info:a, g:0 , f:7 , parinte:None)]
Closed:[]

Pas 2.

S-a extins a.
Open: [(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a)]
Closed:[(info:a, g:0 , f:7 , parinte:None)]

Pas 3.
S-a extins g.
Open: [(info:b, g:10, f:18, parinte:a), (info:c, g:14, f:21, parinte:g), (info:f, g:25:, f:25, parinte:g)]
Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a)]

Pas 4.
S-a extins b.
Open: [(info:d, g:14, f:19, parinte:b), (info:c, g:13, f:20, parinte:b), (info:f, g:25:, f:25, parinte:g)]
//nodul c vechi, din coada open, a fost înlocuit de unul cu cost mai mic
Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a)]

Pas 5.
S-a extins d.
Open: [(info: e, g:16, f:20, parinte:d), (info:c, g:13, f:20, parinte:b), (info:f, g:25:, f:25, parinte:g)]
//cand avem f-uri egale ordonăm descrescător după g (în coada open)
Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a), (info:d, g:14, f:19, parinte:b)]

Pas 6.
S-a extins e.
Open: [(info:c, g:13, f:20, parinte:b), (info:f, g:23:, f:23, parinte:g)] //nodul f vechi a fost inlocuit de unul cu cost mai mic
Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a), (info:d, g:14, f:19, parinte:b), (info: e, g:16, f:20, parinte:d)]

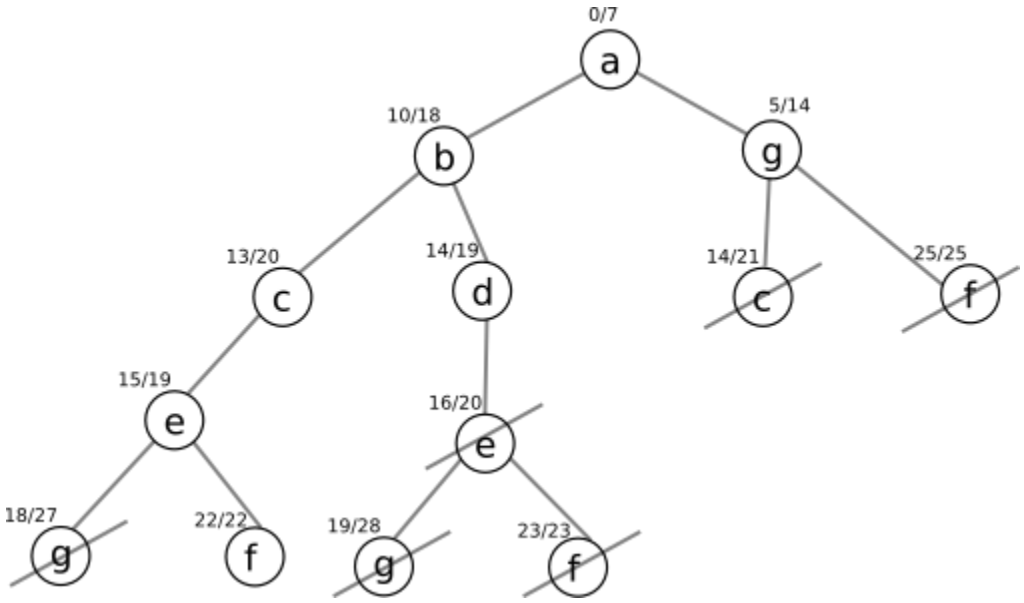
Pas 7.
S-a extins c.
Open: [(info: e, g:15, f:19, parinte:c), (info:f, g:23:, f:23, parinte:g)]
//c-ul se extinde în b si e, dar b-ul a mai fost în drumul curent, conform arborelui, deci nu se adaugă, iar e-ul obtinut e cu cost mai mic decât cel din lista closed; prin urmare, îl ștergem din closed pe e și îl adăugăm în open
Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a), (info:d, g:14, f:19, parinte:b), (info:c, g:13, f:20, parinte:b)]

Pas 8.
S-a extins e.
Open: [(info:f, g:22:, f:22, parinte:g)]
//e-ul se extinde în g si f, dar g-ul obținut e cu cost mai mare decât cel din lista closed, deci nu se adaugă; iar f-ul obținut e cu cost mai mic,deci îl înlocuiește pe cel din lista open.

Closed:[(info:a, g:0 , f:7 , parinte:None),(info:g, g:5, f:14, parinte:a), (info:b, g:10, f:18, parinte:a), (info:d, g:14, f:19, parinte:b), (info:c, g:13, f:20, parinte:b), (info: e, g:15, f:19, parinte:c)]

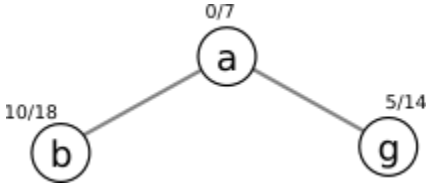
Primul nod din coada open este un nod scop, rezultă că am obținut drumul de cost minim: a->b->c->e->f cu costul 22.

2)
Arborele este:



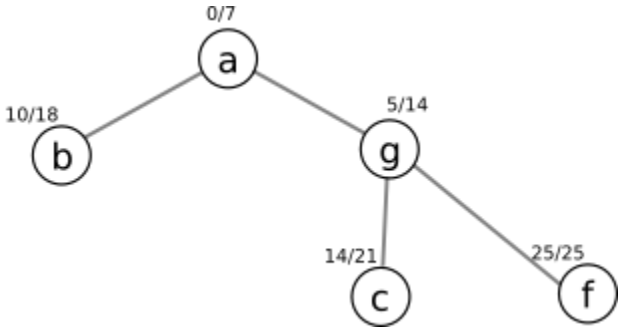
3) Costul arcului e->f să fie modificat în 1; h-ul estimat pentru e, e egal cu 4 și există chiar drumul e->f de cost 1, până la nodul scop. Cum 4>1 reiesel că euristica nu mai e admisibilă.

4)
Limita: 7



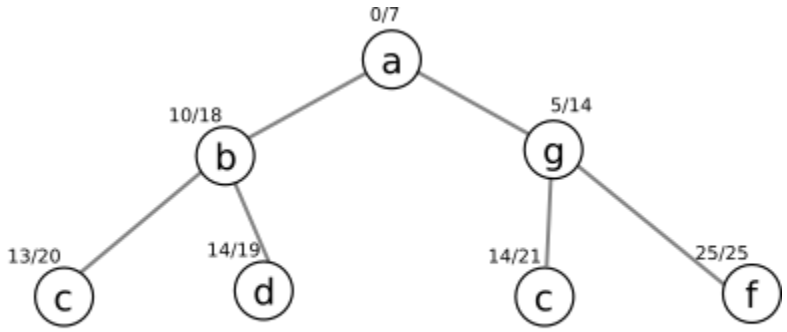
Stiva:
[a]
[a,b]
<- intoarcere
[a,g]

Limita 14:



Stiva:
[a]
[a,b]
<- intoarcere
[a,g]
[a,g,c]
<- intoarcere
[a,g,f]

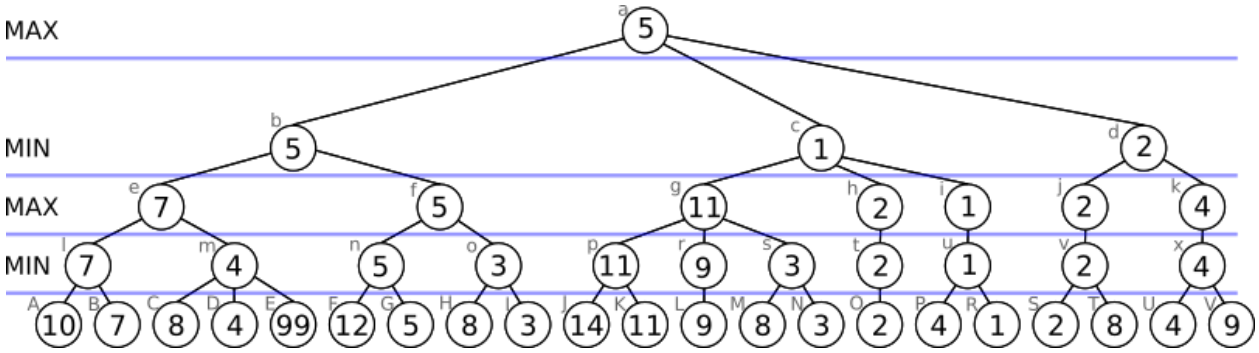
Limita 18:



Stiva:
[a]
[a,b]
[a,b,c]
<- intoarcere
[a,b,d]
<- intoarcere
<- intoarcere
[a,g]
[a,g,c]
<- intoarcere
[a,g,f]

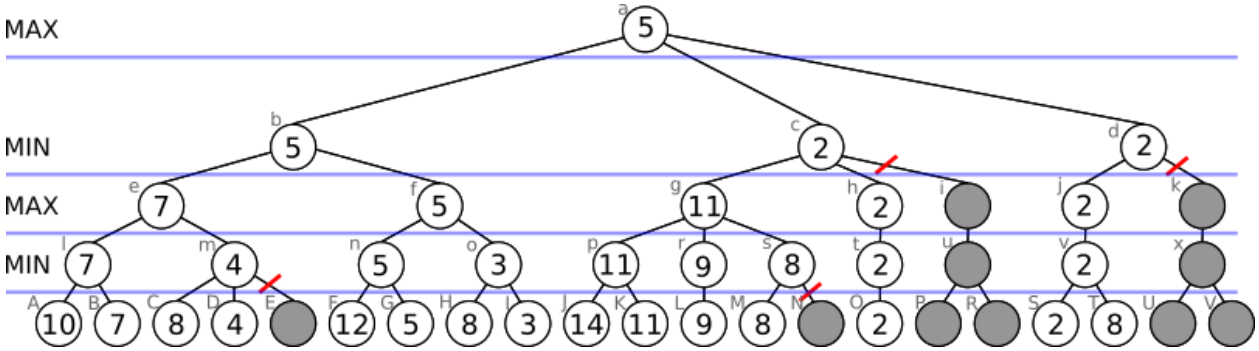
II)

1)



Variatia principală este a-b-f-n-G cu valoarea jocului 5.

2)



După generarea nodului D cu valoarea 4, MAX are de ales între 7 și o valoare mai mică sau egală cu 4, pentru nodul e, prin urmare frații lui D nu mai trebuie calculați

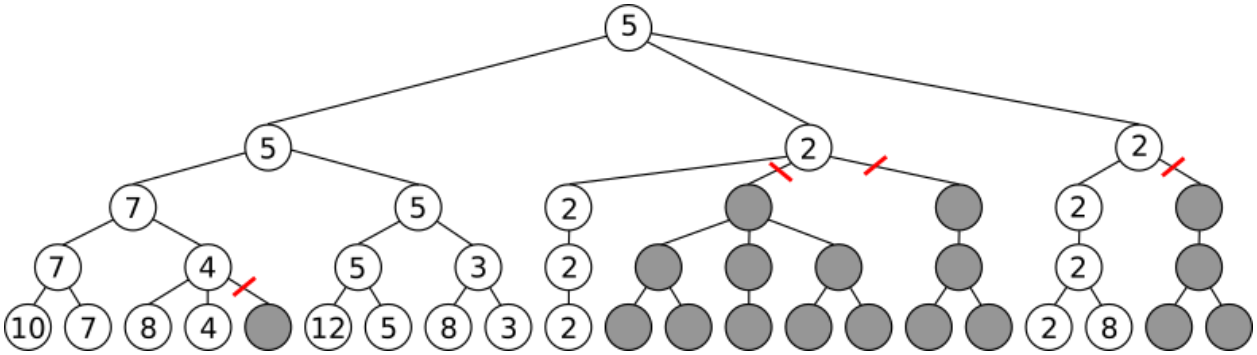
După generarea nodului M cu valoarea 8, MAX are de ales între 11 și o valoare mai mică sau egală cu 8, pentru nodul s, prin urmare frații lui M nu mai trebuie calculați

După evaluarea nodului h la valoarea 2, MAX are de ales între 5 și o valoare mai mică sau egală cu 2, pentru nodul a, prin urmare frații lui c nu mai trebuie calculați

După evaluarea nodului j la valoarea 2, MAX are de ales între 5 și o valoare mai mică sau egală cu 2, pentru nodul a, prin urmare frații lui j nu mai trebuie calculați

3) Reordonare:

Observatie: aici initial uitasem sa marchez nodurile care se taie după reordonare; am corectat:



4)