Cozma Laura-Elena
244

Examen

I Lista open va conține acele noduri care nu au fost încă expandate, iar în fiecare moment va fi sortată crescător după f și descrescător după g.
Lista closed conține nodurile interioare ale arborelui de căutare.

I Inițializare:
Lista open inițială: [(a, g=0, f=inf, p=clone)]
Lista closed inițială: []

II extragem (a, g=0, f=inf, p=clone) din lista open și îl plasăm în lista closed.
Nodul nu e scop deci îl expandăm.
Succesorii săi sunt:
(c, g=8, f=16, p=a)
(g, g=10, f=16, p=a)
(d, g=5, f=17, p=a)
Lista open este:
[(g, g=10, f=16, p=a), (c, g=8, f=16, p=a),
(d, g=5, f=17, p=a)]
Lista closed:
[(a, g=0, f=inf, p=clone)]

III extragem g
el nu e nod scop. Succesorii sunt:
(h, g=13, f=28, p=g)
(f, g=23, f=23, p=g)
(e, g=18, f=22, p=g)
Lista open devine:
[(c, g=8, f=16, p=a), (d, g=5, f=17, p=a),
(e, g=18, f=22, p=g), (f, g=23, f=23, p=g), (h, g=13, f=28, p=g)]

Lista closed:

[(a, g=0, f=inf, p=none), [g, g=10, f=16, p=a)]

IV Extragem c

este nod scop

Succesorii

(h, g=18, f=33, p=c)

(g, g=11, f=17, p=c)

h era în open, dar noul f e mai mare => nu îl adăugăm

g e în closed, noul f e mai mare, nu îl adăugăm

Open: [(d, g=5, f=17, p=a), (e, g=18, f=22, p=g),

(f, g=23, f=23, p=g), (h, g=13, f=28, p=g)]

Closed: [(a, g=0, f=inf, p=none], [g, g=10, f=16, p=a),

(c, g=8, f=16, p=a)]

V Extragem d

este nod scop.

Succesori:

(c, g=7, f=15, p=d)

(h, g=7, f=22, p=d)

(f, g=35, f=35, p=d)

c e în closed, dar cu f mai mare, îl adaug în open.

f_h nou < f_h vechi => adaug în open.

f_f nou > f_f vechi => nu îl adaug

Open: [(c, g=7, f=15, p=d), (e, g=18, f=22, p=g),

(h, g=7, f=22, p=d), (f, g=23, f=23, p=g)]

Closed: [(a, g=0, f=inf, p=none), [g, g=10, f=16, p=a),

[d, g=5, f=17, p=a)]

VI Extragem c

cel e mod scop

(b, g=17, f=32, p=c) nu s schimbă nimic

(g, g=10, f=16, p=c)

Lista open devine:

[(e, g=18, f=22, p=g), (b, g=7, f=22, p=d),

(f, g=23, f=23, p=g)]

Closed: [(a, g=0, f=inf, p=None), [g, g=10, f=16, p=a),

(d, g=5, f=17, p=a), (c, g=7, f=15, p=d)]

VII extragem e

cel e mod scop

(g, g=29, f=35, p=e) nu s adaugă, e pe drum de

la rād la frunză

(f, g=24, f=24, p=e) nu s modifică

Open:

[(b, g=7, f=22, p=d), (f, g=23, f=23, p=g)]

Closed:

[(a, g=0, f=inf, p=None), [g, g=10, f=16, p=a),

(d, g=5, f=17, p=a), (c, g=7, f=15, p=d), (e, g=18,

f=22, p=g)]

VIII extragem b

cel e scop

Succesori:

(d, g=11, f=23, p=b) e pe drumul de la rād.

(f, g=32, f=32, p=b) nu s actualizează

(g, g=11, f=17, p=b) e în closed, f e mai mare,

nu s actualizează;

Open: [(f, g=23, f=23, p=g)]

Closed: [(a, g=0, f=inf, p=None), (g, g=10, f=16, p=a),

(d, g=5, f=17, p=a), (c, g=7, f=15, p=d), (e, g=18, f=22,

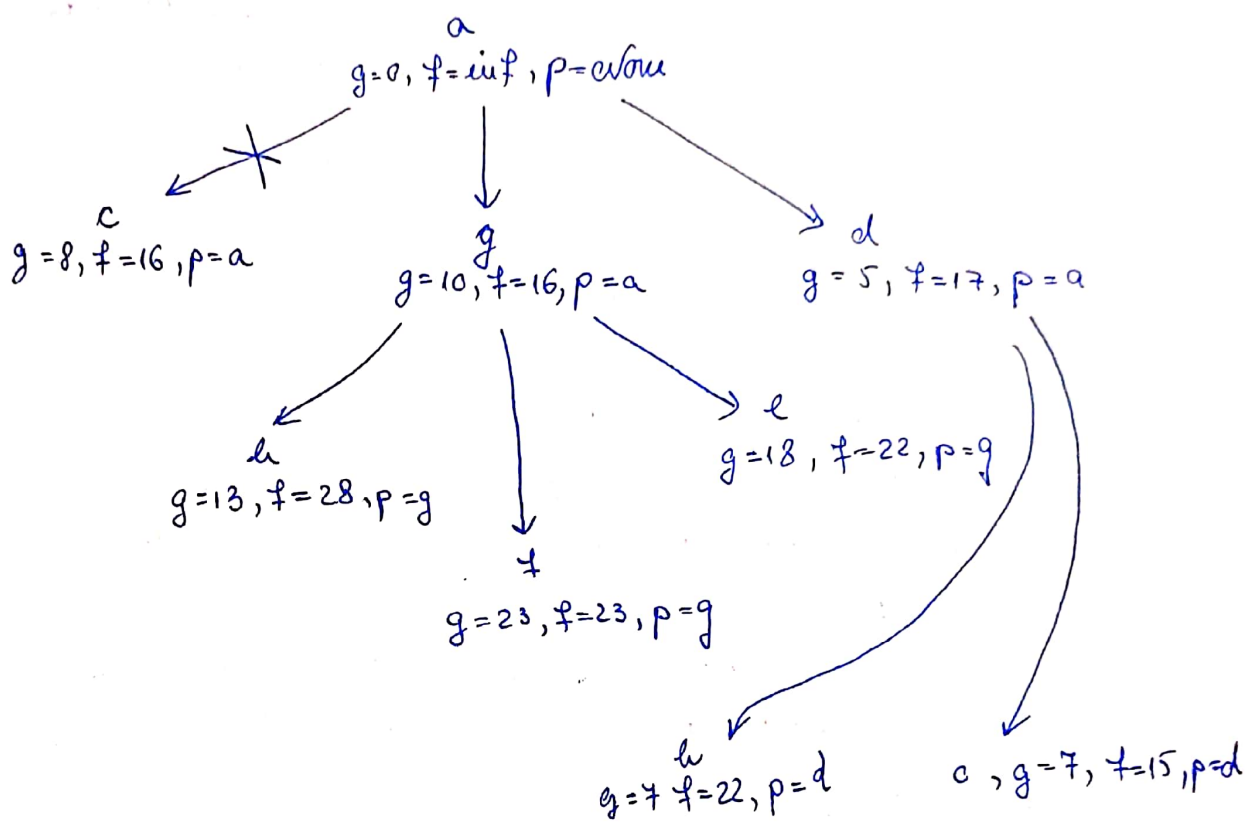
p=g), (b, g=7, f=22, p=d)]

IV Extragem f, e nod scop, nu operăm.

Drumul de cost minim este:

[(a, g = 0, f = inf, p = cărui), (g, g = 10, f = 16, p = a),
(f, g = 23, f = 23, p = g)]

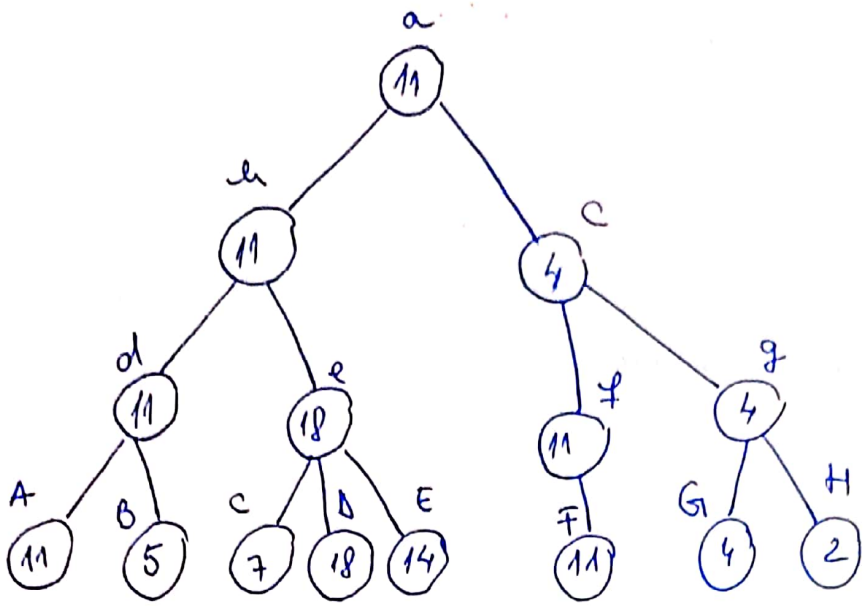
Iar costul minim e 23.



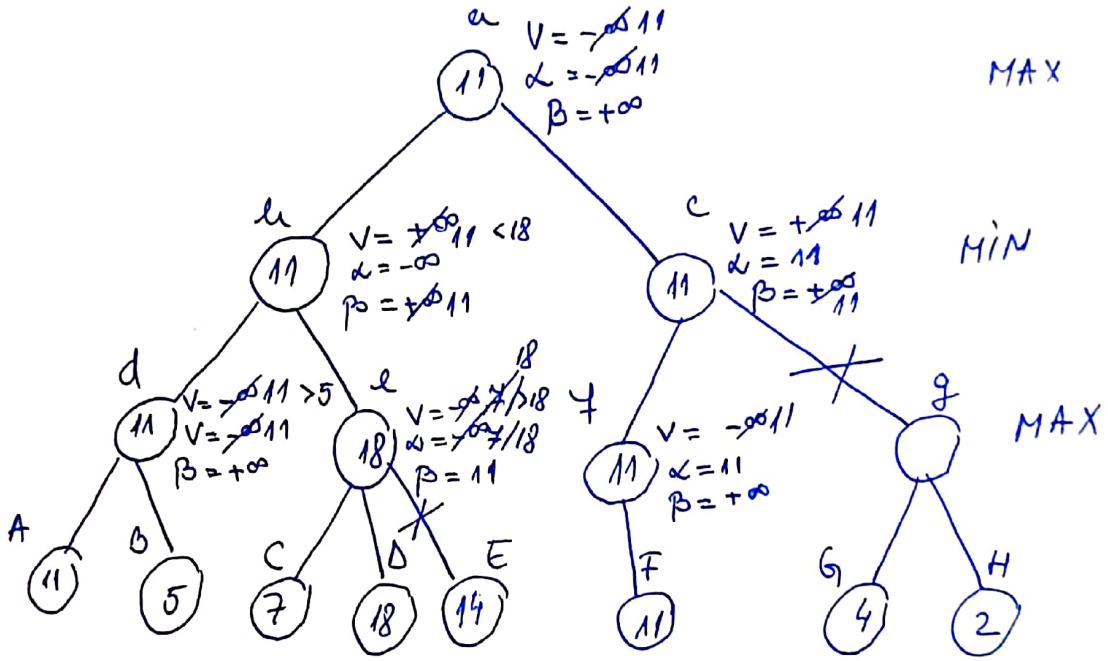
Subiectul 2:

a) a e de tip MAX
b, c MIN
d, e, f, g MAX
A B C D E F G H

Frunze, li se calculează ~~inutil~~ valoarea direct



b)



Retezare e - Ē:

V = 7

Valoarea din D e 18. Cum 7 > 18, actualizăm V. Cum e este nod de tip max, verificăm dacă α < 18 și actualizăm

α = 18

β = 11

Tatăl lui e a impus limita maximă 11 și cum α > β, retezăm ramura e-E pentru că deja e are o valoare mai mare ca 11.

Retezare c - g:

c verifică valoarea din f

$V_f = 11$ $\quad 11 < +\infty \Rightarrow$ se actualizează $V_c = 11$

$V_c = +\infty$

Cum c e nod e tip MIN, verificăm:

$\beta = +\infty < 11 \Rightarrow$ da, stânga

$\alpha = 11$ $\quad \alpha \leq \beta \Rightarrow$ retragem, deoarece a vrea valori mai mari

$\beta = 11$

de 11, iar c îi poate oferi maximu

Deci, max va avea scorul 11