



Pasul 1) Lista open initiala: $[(a, h=\text{inf}), \text{parinte}=\text{None}, f=\text{inf}, g=0]$

Pasul 2) Lista closed initiala: $[]$

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: $[(a, h=\text{inf}), \text{parinte}=\text{None}, f=\text{inf}, g=0]$

Noduri in lista closed: $[]$

Pasul 4) Extragem $((a, h=\text{inf}), \text{parinte}=\text{None}, f=\text{inf}, g=0)$ din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat $((a, h=\text{inf}), \text{parinte}=\text{None}, f=\text{inf}, g=0)$ sunt

nod: (b, h=9), cost arc:7

nod: (c, h=10), cost arc:4

nod: (d, h=11), cost arc:5

Pasul 7) Procesez succesorii.

Lista open dupa ce au fost adaugati succesorii este: [(b, h=9), parinte=a, f=16, g=7) ((c, h=10), parinte=a, f=14, g=4) ((d, h=11), parinte=a, f=16, g=5)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [(c, h=10), parinte=a, f=14, g=4) ((b, h=9), parinte=a, f=16, g=7) ((d, h=11), parinte=a, f=16, g=5)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [(c, h=10), parinte=a, f=14, g=4) ((b, h=9), parinte=a, f=16, g=7) ((d, h=11), parinte=a, f=16, g=5)]

Noduri in lista closed: [(a, h=inf), parinte=None, f=inf, g=0)]

Pasul 4) Extragem ((c, h=10), parinte=a, f=14, g=4) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((c, h=10), parinte=a, f=14, g=4) sunt

nod: (b, h=9), cost arc:2

nod: (e, h=6), cost arc:8

nod: (f, h=10), cost arc:5

Pasul 7) Procesez succesorii.

Nodul ((b, h=9), parinte=a, f=16, g=7) se afla deja in open cu un g estimat (adancime) mai mare, asa ca il actualizez la (redirectionare pointeri):

((b, h=9), parinte=c, f=15, g=6)

Lista open dupa ce au fost adaugati succesorii este: [(b, h=9), parinte=c, f=15, g=6) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [(b, h=9), parinte=c, f=15, g=6) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [((b, h=9), parinte=c, f=15, g=6) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]
Noduri in lista closed: [((a, h=inf), parinte=None, f=inf, g=0) ((c, h=10), parinte=a, f=14, g=4)]

Pasul 4) Extragem ((b, h=9), parinte=c, f=15, g=6) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((b, h=9), parinte=c, f=15, g=6) sunt
nod: (f, h=10), cost arc:6
nod: (g, h=6), cost arc:3

Pasul 7) Procesez succesorii.
Lista open dupa ce au fost adaugati succesorii este: [((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9) ((g, h=6), parinte=b, f=15, g=9)]

Pasul 8) Sortam lista open crescator dupa f:
Lista open dupa sortare: [((g, h=6), parinte=b, f=15, g=9) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [((g, h=6), parinte=b, f=15, g=9) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]
Noduri in lista closed: [((a, h=inf), parinte=None, f=inf, g=0) ((c, h=10), parinte=a, f=14, g=4) ((b, h=9), parinte=c, f=15, g=6)]

Pasul 4) Extragem ((g, h=6), parinte=b, f=15, g=9) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((g, h=6), parinte=b, f=15, g=9) sunt
nod: (i, h=4), cost arc:2
nod: (j, h=3), cost arc:3

Pasul 7) Procesez succesorii.
Lista open dupa ce au fost adaugati succesorii este: [((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9) ((i, h=4), parinte=g, f=15, g=11) ((j, h=3), parinte=g, f=15, g=12)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [(j, h=3), parinte=g, f=15, g=12) ((i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [(j, h=3), parinte=g, f=15, g=12) ((i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]
Noduri in lista closed: [(a, h=inf), parinte=None, f=inf, g=0) (c, h=10), parinte=a, f=14, g=4) (b, h=9), parinte=c, f=15, g=6) (g, h=6), parinte=b, f=15, g=9)]

Pasul 4) Extragem (j, h=3), parinte=g, f=15, g=12) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((j, h=3), parinte=g, f=15, g=12) sunt

Pasul 7) Procesez succesorii.

Lista open dupa ce au fost adaugati succesorii este: [(i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [(i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [(i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]
Noduri in lista closed: [(a, h=inf), parinte=None, f=inf, g=0) (c, h=10), parinte=a, f=14, g=4) (b, h=9), parinte=c, f=15, g=6) (g, h=6), parinte=b, f=15, g=9) (j, h=3), parinte=g, f=15, g=12)]

Pasul 4) Extragem (i, h=4), parinte=g, f=15, g=11) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((i, h=4), parinte=g, f=15, g=11) sunt

Pasul 7) Procesez succesorii.

Lista open dupa ce au fost adaugati succesorii este: [((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [((d, h=11), parinte=a, f=16, g=5) ((e, h=6), parinte=c, f=18, g=12) ((f, h=10), parinte=c, f=19, g=9)]

Noduri in lista closed: [((a, h=inf), parinte=None, f=inf, g=0) ((c, h=10), parinte=a, f=14, g=4) ((b, h=9), parinte=c, f=15, g=6) ((g, h=6), parinte=b, f=15, g=9) ((j, h=3), parinte=g, f=15, g=12) ((i, h=4), parinte=g, f=15, g=11)]

Pasul 4) Extragem ((d, h=11), parinte=a, f=16, g=5) din lista open si il plasam in lista closed

Pasul 5) Nodul nu este scop deci il expandam.

Pasul 6) Succesorii nodului de expandat ((d, h=11), parinte=a, f=16, g=5) sunt

nod: (h, h=0), cost arc:11

nod: (e, h=6), cost arc:5

Pasul 7) Procesez succesorii.

Nodul ((e, h=6), parinte=c, f=18, g=12) se afla deja in open cu un g estimat (adancime) mai mare, asa ca il actualizez la (redirectionare pointeri):

((e, h=6), parinte=d, f=16, g=10)

Lista open dupa ce au fost adaugati succesorii este: [((e, h=6), parinte=d, f=16, g=10) ((f, h=10), parinte=c, f=19, g=9) ((h, h=0), parinte=d, f=16, g=16)]

Pasul 8) Sortam lista open crescator dupa f:

Lista open dupa sortare: [((h, h=0), parinte=d, f=16, g=16) ((e, h=6), parinte=d, f=16, g=10) ((f, h=10), parinte=c, f=19, g=9)]

Pasul 9) Revenim la pasul 3

=====

Pasul 3) Lista open nu e vida, urmeaza pasii repetitivi.

Noduri in lista open: [((h, h=0), parinte=d, f=16, g=16) ((e, h=6), parinte=d, f=16, g=10) ((f, h=10), parinte=c, f=19, g=9)]

Noduri in lista closed: [((a, h=inf), parinte=None, f=inf, g=0) ((c, h=10), parinte=a, f=14, g=4) ((b, h=9), parinte=c, f=15, g=6) ((g, h=6), parinte=b, f=15, g=9) ((j, h=3), parinte=g, f=15, g=12) ((i, h=4), parinte=g, f=15, g=11) ((d, h=11), parinte=a, f=16, g=5)]

Pasul 4) Extragem ((h, h=0), parinte=d, f=16, g=16) din lista open si il plasam in lista closed
Nodul extras din open este nod scop

----- Concluzie -----

Drum de cost minim: [((a, h=inf), parinte=None, f=inf, g=0) ((d, h=11), parinte=a, f=16, g=5) ((h, h=0), parinte=d, f=16, g=16)]

>>>