Algoritmul A*

1. Creează un graf de căutare G, constând numai din nodul iniţial n_0. Plasează n_0 într-o listă numită OPEN.

2. Creează o listă numită CLOSED, care iniţial este vidă.

3. Dacă lista OPEN este vidă, EXIT cu eşec.

4. Selectează primul nod din lista OPEN, înlătură-l din OPEN şi plasează-l în lista CLOSED. Numeşte acest nod *n*.

5. Dacă *n* este un nod scop, opreşte execuţia cu succes. Returnează soluţia obţinută urmând un drum de-a lungul pointerilor de la *n* la n_0 în G. (Pointerii definesc un arbore de căutare şi sunt stabiliţi la pasul 7).

6. Extinde nodul *n*, generând o mulţime, *M*, de succesori ai lui care nu sunt deja strămoşi ai lui *n* în G. Instalează aceşti membri ai lui *M* ca succesori ai lui *n* în G.

7. Stabileşte un pointer către *n* de la fiecare dintre membrii lui *M* care nu se găseau deja în G (adică nu se aflau deja nici în OPEN, nici în CLOSED). Adaugă aceşti membri ai lui *M* listei OPEN. Pentru fiecare membru, *m*, al lui *M*, care se afla deja în OPEN sau în CLOSED, redirecţionează pointerul său către *n*, dacă cel mai bun drum la *m* găsit până în acel moment trece prin *n*. Pentru fiecare membru al lui *M* care se

află deja în lista CLOSED, redirecţionează pointerii fiecăruia dintre descendenţii săi din G astfel încât aceştia să ţintească înapoi de-a lungul celor mai bune drumuri până la aceşti descendenţi, găsite până în acel moment.

8. Reordonează lista OPEN în ordinea valorilor crescătoare ale funcţiei $\hat{f}$. (Eventuale legături între valori minimale ale lui $\hat{f}$ sunt rezolvate în favoarea nodului din arborele de căutare aflat la cea mai mare adâncime).

9. Mergi la pasul 3.

❑