

Laborator 6 (seria 24) (75 puncte)

Important de citit. Convenții.

În subpunctele de mai jos se consideră pașii algoritmului așa cum au fost predați la curs.

Se consideră convenția că MAX este calculatorul și MIN este utilizatorul.

Se consideră adâncimea maximă setată în toate problemele de mai jos ca fiind mai mare sau egală cu 1, unde adâncimea 0 e cea a rădăcinii iar adâncimea 1 e cea a fiilor rădăcinii. Pentru o adâncime maximă setată, ne oprim cu calculul fiilor la acea adâncime (nu vom genera niciun nod la adâncime strict mai mare decât ea).

Se consideră ordinea de evaluare a fraților în arborele Minimax de la stânga la dreapta.

1

Care dintre estimările următoare sunt admisibile pentru problema 8-puzzle. Considerăm costul pe o mutare ca fiind numărul de plăcuțe vecine ale locației din care pleacă plăcuța mutată. De exemplu, dacă plăcuța era în colț și s-a mutat atunci avea 2 vecini (deoarece dintre cele 3 locații vecine una era sigur liberă). (15 puncte)

- ☐ $h(\text{stare}) = \text{numărul de plăcuțe care nu sunt la locul lor}$ ✓
- ☐ $h(\text{stare}) = 2 * (\text{numărul de plăcuțe care nu sunt la locul lor})$ ✓
- ☐ $h(\text{stare}) = 3 * (\text{numărul de plăcuțe care nu sunt la locul lor})$
- ☐ considerăm $d(\text{placuta}) = \text{distanța Manhattan dintre locația curentă a plăcuței și locația sa în starea scop}$.
 $h(\text{stare}) = \text{suma tuturor valorilor } d(\text{placuta}) \text{ pentru toate plăcuțele din stare}$ ✓
- ☐ considerăm $d(\text{placuta}) = \text{distanța Manhattan dintre locația curentă a plăcuței și locația sa în starea scop}$.
 $h(\text{stare}) = 2 * (\text{suma tuturor valorilor } d(\text{placuta}) \text{ pentru toate plăcuțele din stare})$ ✓

2

Care dintre afirmațiile de mai jos sunt adevărate pentru algoritmul Minimax (cu adâncime maximă setată)? (15 puncte)

- ☐ Rădăcina arborelui are întotdeauna un fiu de tip MAX.
- ☐ Fie două noduri n_1 și n_2 frați, atunci fiii celor două noduri sunt ori toți de tip MIN ori toți de tip MAX (de exemplu, nu putem avea un nod de tip MAX fiu al lui n_1 și un nod de tip MIN fiu al lui n_2) ✓
- ☐ Printre frunzele arborelui minimax pot fi și noduri de tip MIN și noduri de tip MAX în același timp ✓
- ☐ Frunzele din arborele minimax sunt întotdeauna stări finale.
- ☐ Orice nod corespunzător unei stări finale a jocului se găsește întotdeauna la adâncime maximă în arborele minimax

3

Care dintre următoarele afirmații sunt adevărate pentru algoritmul Alpha-Beta (cu adâncime maximă setată)? (15 puncte)

- ☐ Pentru aceeași rădăcină (aceeași mutare pentru care calculăm arborii) și aceeași adâncime maximă setată, arborii Minimax are un număr strict mai mare de noduri (în total) decât arborii Alpha-Beta
- ☐ Pentru un nod care nu este retezat, nu putem avea toți fiii lui rețezați ✓
- ☐ Nu se va reteza niciodată un nod care e stare finală
- ☐ Dacă, pentru un nod N am avea în ordinea generării fiii (mutările): f_1, f_2, \dots, f_n , dacă este retezat fiul f_i , cu $i < n$, atunci toți fiii de la f_i până la f_n sunt rețezați. ✓

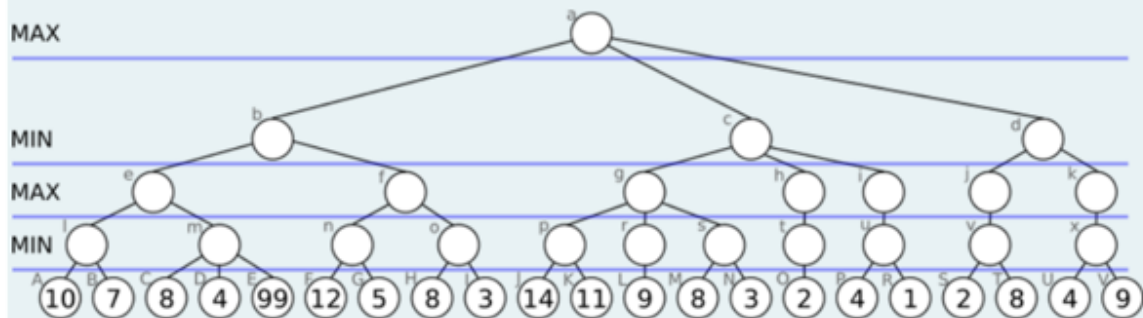
4

Care dintre afirmațiile de mai jos sunt adevărate pentru algoritmi Minimax și Alpha-Beta (cu adâncime maximă setată)? (15 puncte)

- ☐ Valoarea minimax a rădăcinii este sigur una dintre valorile nodurilor aflate la adâncime maximă.
- ☐ Pentru o bună implementare a unui program care joacă un joc, folosind algoritmul minimax, în cadrul arborelui generat de algoritm, valoarea minimax asociată unui nod în care a câștigat calculatorul este mai mică decât valoarea unui nod în care a câștigat jucătorul uman.
- ☐ Eficiența Algoritmului Alpha-Beta depinde de ordinea în care sunt examinați succesorii. ✓
- ☐ Cea mai potrivită valoare pentru o stare de remiză (stare finală în care nici calculatorul și nici jucătorul nu au câștigat) este 0. ✓
- ☐ Pentru un fiu n , al rădăcinii, găsirea unui fiu n_1 al lui n , în care a câștigat MAX, garantează tăierea tuturor fraților lui n aflați în dreapta acestuia.

5

Considerăm arborele Minimax din imagine pentru care cunoaștem valorile din frunze. Care dintre următoarele afirmații sunt adevărate? (15 puncte)



- ☐ Dacă am aplica algoritmul alpha-beta pe acest arbore și valoarea nodului G ar fi 8 în loc de 5, atunci nodul o nu ar mai fi evaluat. ✓
- ☐ În nodul a vom avea valoarea minimax 99.
- ☐ Dacă am aplica algoritmul alpha-beta pe acest arbore, nodurile D și E nu ar mai fi evaluate.
- ☐ Aplicând algoritmul minimax, nodul h va avea în mod sigur valoarea 2. ✓
- ☐ Dacă valorile frunzelor ar fi toate de 10 ori mai mari (pentru orice nod frunză, în loc de valoarea v am avea valoarea $10 \cdot v$) atunci setul de noduri din variația principală nu s-ar schimba. ✓