

1. Care dintre următoarele fraze sunt adevărate pentru o problemă de căutare oarecare, abstractizată la un graf cu muchii ponderate (cu ponderi pozitive, nenule)?

Considerăm lungimea unui drum ca fiind numărul de arce/muchii din acel drum.

Pentru algoritmul A* se consideră implementarea descrisă pe pași de la curs.

(15 puncte)

- ☐ Pentru o evaluare euristică admisibilă, algoritmul A* oferă ca soluție drumul cu numărul minim de noduri, de la nodul start la nodul scop.
- ☐ Algoritmul A* oferă întotdeauna ca soluție drumul de lungime minimă indiferent de costurile de pe muchii/arce și evaluarea euristică asociată nodurilor grafului.
- ☐ Algoritmul A* oferă întotdeauna ca soluție drumul de cost minim, indiferent de costurile de pe muchii/arce și evaluarea euristică asociată nodurilor grafului.
- ☐ Algoritmul A* oferă întotdeauna ca soluție drumul de cost minim indiferent de costurile de pe muchii/arce dacă evaluarea euristică pentru orice nod N este mai mică decât costul oricărui drum de la N la un nod scop. ✓
- ☐ Algoritmul A* oferă întotdeauna ca soluție drumul de cost minim indiferent de costurile de pe muchii/arce dacă evaluarea euristică pentru orice nod N este mai mare decât costul oricărui drum de la N la un nod scop.

2. Care dintre următoarele fraze sunt adevărate pentru orice problemă de căutare, abstractizată la un graf cu muchii ponderate (cu ponderi pozitive, nenule)? (15 puncte)

- ☐ Algoritmul A* poate fi folosit pentru calcularea drumului de lungime maximă prin înmulțirea costului de pe fiecare arc cu -1, lăsând factorul euristic la fel. În acest caz se ia primul drum returnat de algoritmul modificat, considerând costul său ca fiind cel calculat de A*-modificat înmulțit cu -1.
- ☐ Algoritmul A* poate fi folosit pentru calcularea drumului de lungime maximă, ordonând coada OPEN descrescător după priorități și luând mereu primul nod din coadă pentru evaluare.
- ☐ În drumul returnat de A* nu pot exista mai multe noduri scop. ✓
- ☐ Algoritmul A* poate fi folosit pentru calcularea drumului de lungime maximă prin înlocuirea costului c al fiecărui arc cu $1/c$, modificând și factorul euristic h pentru fiecare nod, astfel încât să fie $1/h$. În acest caz se ia primul drum returnat de algoritmul modificat, considerând costul său ca fiind cel calculat însumând costurile inițiale de pe arce/muchii.

3. Care dintre următoarele fraze referitoare la notațiile și formulele folosite în A* sunt adevărate? Notăm cu n un nod oarecare din graf. (15 puncte)

- ☐ $\hat{g}(n)$ este factorul euristic, $\hat{h}(n)$ este factorul de adâncime, iar funcția euristică de evaluare are formula $\hat{h}(n) = -\hat{g}(n) + \hat{f}(n)$
- ☐ $\hat{g}(n)$ este factorul de adâncime, $\hat{h}(n)$ este factorul euristic, iar funcția euristică de evaluare, $\hat{f}(n)$ este suma dintre $\hat{g}(n)$ și $\hat{h}(n)$ ✓
- ☐ Pentru o estimatie $\hat{h}(n) \leq h(n)$ pentru orice nod n din componenta conexă a nodului start, A* returnează în mod cert drumul de cost minim. ✓
- ☐ O estimatie $\hat{h}(n)$ este considerată neadmisibilă, dacă $\hat{h}(n) \leq h(n)$
- ☐ O estimatie $\hat{h}(n)$ este considerată neadmisibilă, dacă $\hat{h}(n) > h(n)$ ✓

4. Care dintre următoarele fraze sunt adevărate? (15 puncte)

- ☐ A* e o tehnică de căutare neinformată.
- ☐ A* e o tehnică de căutare informată. ✓
- ☐ A* se poate aplica numai pe grafuri orientate.
- ☐ A* se poate aplica numai pe grafuri conexe.
- ☐ În cazul în care aplicăm A* pe un graf fără ponderi pe muchii, dacă considerăm costul unei muchii egal cu 1, și estimatia nodurilor este admisibilă, atunci A* calculează drumul de lungime minimă (dacă acesta există) între nodul start și unul dintre nodurile scop. ✓

5. Care dintre următoarele fraze sunt adevărate?

Observatie (necesară pentru unele subpuncte): Vom presupune că nodurile de cost egal sunt evaluate mereu după aceeași relație de ordine pe stări. De exemplu, dacă relația de ordine pe stări este cea alfabetică, dacă în coadă avem nodurile b și c ambele cu costul estimat \hat{f} egal cu 5, b va fi înaintea lui c. De asemenea, presupunem că estimațiile pentru graful de dinaintea de eventuala modificare (din fiecare subpunct) sunt admisibile. (15 puncte)

☐ Considerăm un graf neorientat oarecare cu un nod start a și un singur nod scop z, iar soluția returnată de A^* este $(a, n_1, n_2, \dots, n_k, z)$, unde n_1, n_2, \dots, n_k sunt noduri din graf. Dacă schimbăm problema, considerând nodul start, ca fiind z, și unicul nod scop ca fiind a, și modificând estimațiile \hat{f} astfel încât să fie în continuare admisibile, atunci drumul returnat de A^* este întotdeauna $(z, n_k, \dots, n_2, n_1, a)$.

☐ Considerăm un graf neorientat oarecare cu un nod start a și un singur nod scop z, iar soluția returnată de A^* este $(a, n_1, n_2, \dots, n_k, z)$, unde n_1, n_2, \dots, n_k sunt noduri din graf. Dacă schimbăm problema, considerând nodul start ca fiind z, și unicul nod scop ca fiind n_i , unde n_i este unul dintre nodurile n_1, \dots, n_k , și modificând estimațiile \hat{f} astfel încât să fie în continuare admisibile, atunci drumul returnat de A^* este întotdeauna (z, n_k, \dots, n_i) .

☐ Considerăm un graf neorientat cu un nod start a și cu mai multe noduri scop dintre care unul este z, iar soluția returnată de A^* este $(a, n_1, n_2, \dots, n_k, z)$, unde n_1, n_2, \dots, n_k sunt noduri din graf. În acest caz putem spune cu certitudine că niciunul dintre nodurile n_1, n_2, \dots, n_k nu este nod scop. ✓

☐ Considerăm un graf neorientat oarecare cu un nod start a și un singur nod scop z, iar soluția returnată de A^* este $d = (a, n_1, n_2, \dots, n_k, z)$, unde n_1, n_2, \dots, n_k sunt noduri din graf. Dacă schimbăm problema, considerând nodul start, ca fiind z, și unicul nod scop ca fiind a, și modificând estimațiile \hat{f} astfel încât să fie în continuare admisibile, atunci drumul returnat de A^* de la z la a, e întotdeauna de aceeași lungime cu d.

☐ Considerăm un graf neorientat oarecare cu un nod start a și un singur nod scop z, iar soluția returnată de A^* este $d = (a, n_1, n_2, \dots, n_k, z)$, unde n_1, n_2, \dots, n_k sunt noduri din graf. Dacă schimbăm problema, considerând nodul start, ca fiind z, și unicul nod scop ca fiind a, dar lăsăm valorile estimațiilor \hat{f} , atunci drumul returnat de A^* de la z la a, e întotdeauna de aceeași lungime cu d.

6. Considerăm că graful problemei este un graf neorientat de tip arbore cu muchii ponderate și costuri numere naturale mai mari sau egale cu 1.

Vom presupune că nodurile de cost egal sunt evaluate mereu după aceeași relație de ordine pe stări atât în A^* cât și în BreadthFirst.

Adâncimea unui nod n este numărul de muchii aflate în lanțul de la rădăcină la nodul n .

Se presupune că estimăția folosită de A^* este admisibilă.

Bifați afirmațiile adevărate: (15 puncte)

☐ Considerăm că nodul start e frunză și nodurile scop sunt de asemenea frunze (dar diferite de nodul start). Pentru fiecare legătură (muchie) $m=(n_1,n_2)$ din graf, alegem o funcție de cost $cost(m)=k \cdot \min_a + 1$, unde \min_a reprezintă minimul dintre adâncimile nodurilor și $k > 2$. În această situație, primul drum soluție returnat de A^* este și primul drum soluție returnat de BreadthFirst.

☐ Considerăm că nodul start e rădăcina și nodurile scop sunt frunze. Pentru fiecare legătură (muchie) $m=(n_1,n_2)$ din graf, alegem o funcție de cost $cost(m)=k \cdot \min_a + 1$, unde \min_a reprezintă minimul dintre adâncimile nodurilor și $k > 2$. În această situație, primul drum soluție returnat de A^* este și primul drum soluție returnat de BreadthFirst. ✓

☐ Pentru fiecare legătură (muchie) $m=(n_1,n_2)$ din graf, alegem o funcție de cost $cost(m)=1$, atunci indiferent care este nodul start și care sunt nodurile scop, primul drum soluție returnat de A^* este și primul drum soluție returnat de BreadthFirst (în cazul în care există un drum soluție). ✓

☐ Considerăm că nodul start e frunză și nodurile scop sunt de asemenea frunze (dar diferite de nodul start). Pentru fiecare legătură (muchie) $m=(n_1,n_2)$ din graf, alegem o funcție de cost $cost(m)=ad_max - \max_a + 1$, unde \max_a reprezintă maximul dintre adâncimile nodurilor n_1 și n_2 , iar ad_max este adâncime maximă a grafului. În această situație, primul drum soluție returnat de A^* este și primul drum soluție returnat de BreadthFirst.

7. Considerăm că graful problemei este un graf oarecare, neorientat, de tip arbore având muchii cu ponderi numere naturale mai mari strict decât 1. (15 puncte)

☐ Dacă și nodul start și nodurile scop sunt frunze (diferite de nodul start), atunci, întotdeauna, nodul rădăcină din graful problemei va fi evaluat și expandat exact o singură dată de algoritmul A^* până la oferirea soluției.

☐ Dacă și nodul start și nodurile scop sunt frunze (diferite de nodul start), atunci atât părintele nodului start, cât și părintele nodului scop (din soluție) vor fi întotdeauna evaluate și expandate exact o singură dată de algoritmul A^* până la oferirea soluției. ✓

☐ Dacă și nodul start și nodurile scop sunt frunze (diferite de nodul start), atunci fiecare nod existent în graful problemei va fi evaluat și expandat exact o singură dată de algoritmul A^* până la oferirea soluției.

☐ Dacă nodul start este rădăcina și nodurile scop sunt frunze, drumul returnat de A^* întotdeauna se va termina cu nodul scop aflat la cea mai mică adâncime.