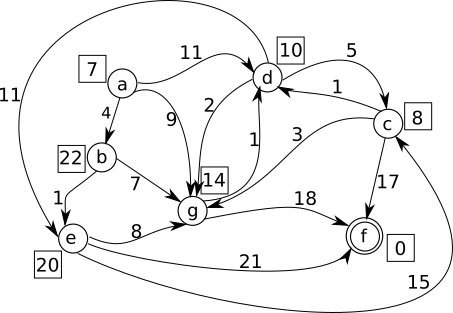
Seminar A\*

**- Seminariile 1 și 2 -**

I) Rezolvați subpunctele de mai jos pentru următorul graf (*a* e **nodul start** și *f* e **nodul scop**). Costurile arcelor sunt trecute în dreptul lor, iar estimațiile sunt trecute în pătrățele, lângă noduri. Considerăm un drum soluție ca fiind un drum de la nodul start la un nod scop.



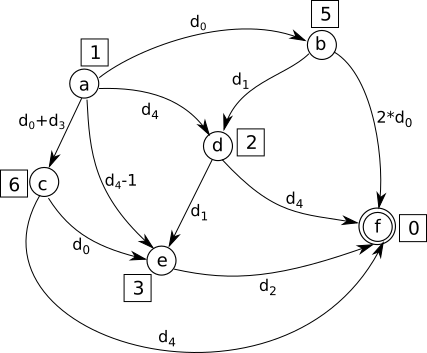
1. Care este drumul soluție de lungime minimă? Este acesta și drumul de cost minim?
2. Ce este o euristică consistentă? Dar admisibilă? Euristica dată este admisibilă? Dar consistentă?
3. Să se realizeze arborele A\* și să se determine drumul de cost minim. Arătați la fiecare pas cum se modifică listele OPEN și CLOSED
4. Considerăm și nodul e ca fiind nod scop, dar pastram euristicile. Care este primul drum returnat de A\*? (trebuie să refaceți arborele?) Mai este euristica admisibilă? Dacă nu, corectați euristica. Pentru o euristică neadmisibilă mai avem garantat faptul că A\* returnează drumul de cost minim?
5. Presupunănd că am schimba euristica nodului d din 10 în 21. Mai este aceasta o euristică admisibilă?
6. Este vreodată folosită euristica nodului start? Justificați.
7. Pentru o euristică admisibilă ce valori avem mereu pentru nodurile scop?
8. Cum ar trebui să schimbăm euristica dacă avem și e și c considerate noduri speciale, și scopul e să găsim un drum care trece prin ambele, intai prin e si apoi prin c (nu mai e obligatoriu să ajungă în f)? În acest caz, drumul de cost minim ar conține arcul e->c? Dar dacă trebuie doar să trecem prin ele, indiferent de ordine? Dar dacă scopul e să trecem prin k noduri date indiferent de ordine?
9. Dați exemplu de euristică astfel încât numărul de extrageri de nod din coada OPEN realizate de A\* pe acest graf să fie egal cu numărul de noduri din drumul de cost minim. Putem avea mai puține extrageri din coada OPEN?
10. Cum s-ar schimba problema dacă scopul nostru ar fi să găsim un drum cu un arc scop (drumul soluție trebuie obligatoriu să treacă prin acel arc); de exemplu, arcul e->c? Găsiți un exemplu de euristică admisibilă. Dar dacă se cere un drumul soluție de cost minim, care trece întâi prin arcul e->c și apoi prin d->g? Dar dacă scopul e să trecem prin cele două arce indiferent de ordinea parcurgerii? Dar dacă scopul e să trecem prin k arce date indiferent de ordine?

242, 241: facut pana la exercitiul 7 inclusiv

II) Întrebări generale despre A\*

1. Care este drumul de cost minim dacă nodul start e și scop și cât e costul?
2. Poate drumul returnat de A\* să treacă prin două noduri scop? De ce?
3. Pentru toate intrebarile consideram un algoritm alg\_h care ne da euristici admisibile pentru toate nodurile. Fie un arbore avand nodul start chiar radacina si un singur nod scop. Este necesar A\* sau putem gasi o metoda mai eficienta? Dar daca nodul start e cea mai din stanga frunza iar nodul scop cea mai din dreapta frunza in cadrul unui arbore binar complet? Aceeasi intrebare pentru arbore oarecare. Dar daca e vorba de un nod start si un nod scop oarecare in arbore?
4. În cazul în care avem de ales dintre două noduri cu valoarea f egală, putem impune o alegere prioritară?
5. Fie un graf oarecare și o funcție numită scop(nod) care are valoare 1 pentru nod scop și 0 pentru celelalte noduri. Considerăm cel mai mic cost (de arc) din tot graful notat cu *c\_min*. Construiți o euristică admisibilă pentru acest graf. E și consistentă? Construiți și o euristică neadmisibilă.
6. Aceeași problemă cu cea de mai sus, dar în care cunoaștem doar *c\_mediu* (media aritmetică a tuturor costurilor arcelor din graf). Idem pentru *c\_max* (cel mai mare cost de arc din tot graful).
7. O euristică rămâne admisibilă orice nod am alege din graf ca fiind nodul de start? Justificați.
8. Fie 3 euristici admisibile ĥ1, ĥ2, ĥ3 (dar diferite două câte două). Cum se pot combina acestea într-o euristică admisibilă ĥ, mai bună decât toate cele 3?
9. Fie un graf și k noduri scop. Pentru fiecare nod scop *i* în parte cunoaștem o estimație admisibila ĥi care nu e neapărat admisibilă pentru restul nodurilor scop. Găsiți o formulă care implică toate euristicile pentru a calcula o euristică admisibilă pentru toate nodurile scop.
10. Dacă am considera o euristică admisibilă ĥ și am construi o altă euristică ĥ1(nod)=ĥ(nod)/2 pentru orice nod din graf, mai rămâne euristica admisibilă? Dar dacă ĥ1(nod)=ĥ(nod) \* 2? Ce alte operații putem face asupra lui ĥ ca să obținem o altă euristică admisibilă? Dacă ĥ1(nod)=ĥ(nod)+c unde c este o constantă reală strict pozitivă, putem spune sigur despre ĥ1 că e admisibilă sau neadmisibilă?
11. Dacă am considera o euristică neadmisibilă ĥ și am construi o altă euristică ĥ1(nod)=ĥ(nod)/2 pentru orice nod din graf, mai rămâne euristica neadmisibilă? Dar dacă ĥ1(nod)=ĥ(nod) \* 2? Ce alte operații putem face asupra lui ĥ ca să obținem o altă euristică neadmisibilă?
12. Poate un nod să existe în același timp în listele *OPEN* și *CLOSED*?
13. Care e lungimea minima/maximă a lui OPEN? Dar CLOSED? Presupunem ca un nod e pus in closed inainte de verificarea ca e scop si expandarea care urmeaza imediat in cadrul algoritmului.
14. Dacă toate nodurile scop sunt pe altă componentă conexă decât nodul start, cum vor arăta listele *OPEN* și *CLOSED* la terminarea rulării algoritmului A\*.
15. Cum s-ar putea aplica A\* în cazul în care am avea mai multe noduri start posibile?
16. Ar putea A\* să funcționeze pentru un graf cu costuri negative? De ce?
17. O euristică consistentă este întotdeauna admisibilă? O euristică admisibilă este întotdeauna consistentă?
18. Fie un graf cu o euristică admisibilă asociată. Dacă tăiem un număr k de arce astfel încât în continuare să mai existe un drum de la nodul start la unul dintre nodurile scop, mai rămâne euristica admisibilă?
19. Dacă toate muchiile unui graf au cost 1, drumul de cost minim e și de lungime minimă? Dați exemplu de o euristică admisibilă.
20. Desenați un graf complet în formă de pătrat. Stabiliți costurile și două euristici admisibile ĥ1 și ĥ2 astfel încât să existe două drumuri de cost minim, iar, pentru ĥ1, A\* să returneze primul drum de cost minim și pentru ĥ2 celălalt.
21. Poate fi A\* folosit pentru a determina drumul de cost maxim doar schimbând în algoritm faptul că ordonăm în OPEN nodurile în mod descrescător după cost?
22. Fie o stare în cadrul algoritmului în care toate nodurile grafului sunt în CLOSED. Ce puteți spune despre această stare și/sau despre graf?
23. Fie *no(t)* numărul de noduri din OPEN la momentul t și *nc(t)* numărul de noduri din CLOSED la momentul *t*. Putem spune mereu că *no(t1)+nc(t1) < no(t2)+nc(t2)* când *t1<t2*?
24. Fie un graf complet în care nodurile sunt numerotate de la 1 la N. O muchie de la nodul *n1* la nodul *n2* are drept cost *|n2-n1|*. Considerăm nodul 1 ca fiind nod inițial (start) și nodul *K*, cu *1<K≤N* ca fiind nod scop. Dați exemplu de o estimație admisibilă. Cum am putea face estimația cât mai eficientă? Câte drumuri de cost minim există? Aplicând principiul f̂-urilor egale, cine va fi primul drum returnat de A\*?
25. Presupunem ca OPEN=[a->b(15/10), a->c(17/9), a->d->e(18/10)]. Putem spune precis ce noduri sunt in CLOSED?

|||) Se dă graful orientat cu arce ponderate din imagine.



Pentru unele arce ponderile nu sunt precizate, fiind înlocuite de identificatorii *di*. Euristica dată este una admisibilă (estimația pentru fiecare nod e trecută în pătrățelul de lângă nod).

Costurile arcelor sunt **numere naturale nenule**.

Nodul f este nod scop.

Un drum de cost minim alegând ca nod de start pe *a*, este a->b->d->e->f cu costul 8 **dar nu este unicul drum de acest cost**.

Care dintre frazele următoare sunt adevărate în condițiile date ale problemei?

1. d0 și d2 sunt egale
2. d3 este sigur mai mare decât d0
3. Orice număr natural nenul am pune în d4, condițiile problemei se respectă în continuare.
4. d4 e de 2 ori mai mare decât d0
5. dacă înlocuim estimația curentă a lui b cu d1, rămâne în continuare admisibilă
6. d0 este sigur egal cu 4
7. Alegând ca nod de start pe *a*, există în mod cert în graf macar un drum de cost 9.

IV) Răspundeți la următoarele întrebări legate de problema canibalilor și misionarilor:

1. Pentru N=3 și M=2, care dintre următoarele stări nu sunt valide (nu pot fi informații în arborele de drumuri), cu notațiile de la laborator (canibali\_mal\_initial, misionari\_mal\_initial, locatie\_barca): (0,0,0),(2,3,1), (0,0,1), (3,2,1), (3,3,0),(1,2,1), (1,1,1)
2. Dați exemple și de alte reprezentări (eficiente pentru informația nodului). Pot fi similare cu cea de la laborator (dar nu identice).
3. Câte noduri (cu informatii valide) are graful asociat problemei canibalilor și misionarilor pe caz general (cu N misionari, N canibali și M locuri în barcă)
4. Problema are soluție pentru orice M,N numere naturale nenule?
5. Problema are mereu soluție pentru M≥4?

V) Cum s-ar schimba euristica dacă problema canibalilor și misionarilor ar avea una din următoarele modificări:

1. costul fiecărui transport dus este 1 dar întors este 3
2. costul e dat de numarul de oameni din barca
3. costul mutarii unui canibal este 1 si al unui misionar este 3
4. trebuie sa mutam doar misionarii pe celalat mal. Idem daca trebuie sa mutam doar canibalii
5. oamenii trebuie sa se duca pe celalalt mal si sa se si intoarca toti. Cum s-ar schimba starea in cazul acesta?

VI) Fie un graf format din cuvinte (cu același număr de litere, NL) din dictionar. Doua cuvinte sunt legate intre ele printr-o muchie daca diferă prin exact o literă.

Exemple:

masa->casa->case->care->oare

ceapa->teapa->trapa->trupa->trusa->truse

Dați un exemplu de euristică pentru:

1. costul unei muchii egal cu 1
2. costul unei muchii e dat de 1+poz, unde poz e pozitia literei schimbate
3. costul unei muchii este egal cu numarul de ordine al literei care urmeaza sa se schimbe (pentru *a* e 1, pentru *b* e 2 *etc)*
4. costul unei muchii egal cu numarul de ordine al literei in care s-a schimbat litera de pe aceeasi pozitie din cuvantul anterior (pentru *a* e 1, pentru *b* e 2 *etc)*

VII) Fie un joc de puzzle 2D care se desfășoară pe o tablă pătrată N x N. Fiecare celulă a tablei poate conține o piesă de o anumită culoare sau poate fi goală. Obiectivul jocului este de a rearanja piesele astfel încât toate piesele de aceeași culoare să formeze un bloc compact, adică să fie adiacente pe orizontală sau verticală.

Jucătorul poate efectua următoarele mutări:

* Mutarea unei piese dintr-o celulă în altă celulă goală, cu cost N. **Se discută separat cazurile în care mutăm piesa doar pe o celulă vecină sau oriunde în grid.**
* Mutarea unui grup de piese adiacente pe orizontală sau verticală cu un singur pas, cu cost 1 (totusi in urma mutarii, acestea nu au voie sa se suprapuna cu alte piese de alta culoare)

1. Descrieți o structură de date adecvată pentru a reprezenta starea jocului, inclusiv configurația pieselor, pozițiile lor și informațiile necesare pentru a efectua mutări.
2. Creați o euristică h(n) care să estimeze costul minim necesar pentru a ajunge la starea finală a jocului. Demonstrați că euristica voastră este admisibilă.

VIII) **Problema lacatului**

**Context.**

Se considera un lacat cu un numar N de incuietori legate intre ele, puse una in continuarea celeilalte astfel incat cu o cheie speciala se pot accesa toate incuietorile in acelasi timp si se pot incuia/descuia dupa caz. O cheie e impartita pe un numar de zone egal cu numarul de incuietori. O astfel de zona poate incuia descuia sau lasa in aceeasi stare o incuietoare, independent de efectul celorlate portiuni ale cheii asupra celorlalte incuietori. O cheie e codificata printr-o secventa de litere din multimea {d,g,i}, unde d inseamna ca are rolul de a descuia, i are rolul de a incuia iar g e gol, efect nul, lasa incuietoarea corespunzatoare asa cum a gasit-o. O cheie poate fi folosită de maxim K ori înainte de a se strica (nu mai poate fi folosită dacă se strică). O incuietoare poate fi de mai multe ori inchisa, adica, daca a fost incuiata de N ori va trebui sa fie descuiata de N ori ca sa fie deschisa. O incuietoare deja descuiata daca e deschisa de cheie va ramane tot descuiata.  
De exemplu daca avem 6 incuietori si primele trei sunt inchise (o data), celelalte 3 deschise si aplicam cheia dgidgi

| i1 | i1 | i1 | d | d | d |
| --- | --- | --- | --- | --- | --- |
| d | g | i | d | g | i |
| d | i1 | i2 | d | d | i1 |

**Initial toate incuietorile sunt inchise o data**

#### **Costul**

Costul va fi egal cu totalul descuierilor pe toate încuietorile (reamintim ca o încuietoare e una dintre părțile lacătului). O descuiere e fie o decrementare a numărului de încuieri, fie o trecere de la o încuiere la "deschis". Dacă încuietoarea e una cu truc și e încuiata ca efect al descuierii altei încuietori, dar în același timp pe poziția ei cheia are "d", atunci nu se adună la cost (presupunem ca de fapt cheia blochează încuierea)

Exemplu de date de intrare:

K = 3

Chei:

idddgii

idiidid

iidddii

iddgiig

dgggiid

didiigg

gigddgg

gdggggg

gggiggd

dgggddd

IX) Gasiti euristici pentru problema blocurilor modificata astfel:

a) consideram ca o stare este scop dacă toate stivele din ea sunt de înălțimi egale

b) consideram ca o stare este scop dacă toate blocurile sunt pe o singura stiva (nu conteaza care)

c) consideram ca o stare este scop dacă orice stiva cu blocuri are imediat in stanga si imediat in dreapta cate o stiva vida si nu avem stive vide alaturate

d) consideram ca simbolurile de pe blocuri se pot repeta si vrem cate o coloana cuprinzand blocuri doar cu acelasi simbol

e) blocurile au valori numerice. Consideram ca o stare este scop dacă toate stivele au ca valori doar blocuri care au cel mai mare divizor comun diferit de 1 (cmmdc(blocurile de pe stiva) > 1).

Exemplu de stare initiala:

[[2,10,55,14],[27, 8, 50],[3]]

Exemplu de stare finala:

[[2,14,8, 50],[55,10],[3,27]

Exemplu de stare initiala fara solutie: [[3,7],[5,11],[17]]

f) blocurile au valori numerice. Scopul e sa ajungem cu stivele cu sume egale.

g) blocurile au valori numerice. Scopul e sa ajungem cu stivele ordonate crescator (de la varf spre baza).