Subiecte examen

1. Sa se spuna ce algoritm facut la curs poate sa determine eficient care este drumul de lungime maxima intr-un graf aciclic orientat cu lungimi pe arce si sa se schiteze idea folosirii acestuia si modificarile necesare pentru rezolvare.
2. Sa se spuna ce schema de algoritm facuta la curs poate sa determine, cat mai eficient posibil, daca un careu de rebus are o asignare posibila de cuvinte (nu se considera definitiile cuvintelor). Sa se schiteze idea rezolvarii. Se presupune ca exista un dictionar in care cuvintele sunt reprezentate prin vectori de caractere. (Un careu de rebus este o matrice m x n cu unele cellule goale. Toate celelalte celule trebuie sa primeasca cate o litera astfel incat secventele maximale de celule pe orizontala si verticala sa fie cuvinte valide).
3. Comparati A\* si parcurgerea in adancime.
4. Explicati pe un exemplu simplu de ce algoritmul Dijkstra merge doar pentru costuri pozitive pe arce.
5. Fie un graf neorientat cu n noduri si a arce. Care este numarul minim si maxim de punti si puncte de articulatie ale acestui graf? Justificati. Dar daca graful are puncte de articulatie dar nu are punti?
6. Dati 4 modalitati de a incerca imbunatatirea performantei unui algoritm de backtracking.
7. Care sunt principalele asemanari si diferente intre schemel de construire a algoritmilor Greedy si Programare dinamica?
8. Cum se poate modifica Floyd-Warshall ca sa calculeze inchiderea tranzitiva a unui graf?/Sa se modifice un algortim facut la curs ca sa se calculeze inchiderea tranzitiva a unui graf.
9. Care este numarul minim si maxim de CTC intr-un graf cu n noduri?
10. Ce algoritm se obtine daca avem h(n) = 0 la A\*?
11. Care este diferenta intre alg Las Vegas si Monte Carlo?
12. Schitati reteaua de restrictii pentru problema unei asignari de cifre pentru litere astfel incat sa fie valabila adunarea: SEND + MORE = MONEY
13. Specificati 2 euristici pentru problema aranjarii a 8 patrate discutata la curs. Ce puteti spune despre aceste euristici din pcunt de vedere al algoritmului A\*?/Dati exemplu de doua functii euristice care estimeaza numarul de mutari pana la solutie in problema mutarii celor 8 patrate (din 9). Care din ele este mai informata?
14. Sa consideram multimea elevilor de la un bal de absolvire. Pentru ultimul dans al balului, fiecare fata a facut o lista cu colegii cu care i-ar placea sa danseze. Spuneti cum ati putea construi un graf pentru aceasta problema, ce tip de graf este acesta si cu ce algoritm ati putea rezolva eficient problema de a gasi o asignare a perechilor de dans astfel incat sa fie un numar maxim de perechi. Precizam ca aignarea perechilor se face astfel incat orice fata sa nu danseze cu un coleg care nu e pe lista de preferinte.
15. Daca nu ati avea alta varianta de algoritm, cum ati putea folosi schema minimax pentru un program de jucat sah?
16. Enumerati utilizarile folosite in algoritmii de la curs ale timpilor de debut si de finis de la parcurgerea in adancime a unui graf.
17. Dati exemplu de un graf cu n noduri si m > n arce si o parcurgere in adancime care genereaza n arbori de parcurgere a grafului.
18. Cum se poate modifica numarul de componente tare conexe dintr-un graf la adaugarea unei noi muchii? Justificati.
19. Transformand unul din algoritmii facuti la curs, scrieti in pseudocod un algoritm de detectare eficienta a existentei ciclurilor intr-un graf orientat. Justificati. Ce complexitate are?
20. Cum se transforma o problema de determinare a fluxului maxim intr-o retea cu mai multe surse si mai multe destinatii intr-o problema de determinare a fluxului maxim intr-o retea cu o sursa si o destinatie.
21. Modificati un algortim de la curs astfel incat sa detecteze daca un graf e conex sau nu.
22. Mentionati de ce e mai bun AC3 fata de AC1.
23. Dati un exemplu pentru care Ford Fulkelson merge prost. Propuneti o solutie.
24. Care sunt pasii principali care trebuie urmati pentru a rezolva o problema folosind programarea dinamica? Exemplificati pe scurt aplicarea acestor pasi pe una dintre problemele discutate la curs./ Exemplificare pe problema parantezarii matricilor.
25. Scrieti diferentele principale intre backtracking si algoritmii de satisface a constrangerilor (CSP) (ce problema rezolva, care este complexitatea, etc.)? Cum putei combina CSP si backtracking pentru rezolvarea unei probleme dificiel (de ex. Sudoku)?
26. Ce este o muchie inapoi (ce cuolri au varfurile ce determina muchia respectiva)? Demonstrati ca un graf aciclic nu are voie sa contina muchii inapoi in parcurgera in adancime.
27. Schitati pe scurt pseudocodul pentru algorimtul minimax. Spuneti apoi minim 2 posibiliati prin care acesta poate fi imbunatatit pentru a reduce timpul de rezolvare.
28. In ce cazuri complexitatea cautarii drummurilor de cost minim intr-un graf orientat G=(N, A) cu costuri pozitive este O(#N+#A)?
29. Ce este o cale reziduala intr-o retea de conducte si cum se poate obtine? Care este capacitatea reziduala a caii si cum se poate determina? Exemplificati.
30. Comparati A\* si cautarea in latime.
31. Ce tip de algoritmi ati folosi daca ar trebui sa calculati care este unul dintre cele mai mari numere prime?
32. Evidentiati etapele cele mai importante ale strategiei de rezolvare divide et impera, dupa care explicati cum poate fi folosita aceasta pentru a calcula eficient x pow n (x ^ n), pentru x – numar real, iar n >= 0.
33. Explicati utilitatea algoritmilor de satisfacere/propagare a constrangerilor, pornind de la functionalitatea oferita de REVISE. Trecand apoi in revista algoritmul ACI, explicati de ce acesta este considerat ineficient.
34. Definiti notiunea de dominanta pentru algoritmul A\*, explicand de ce aceasta este importanta in alegerea euristicii. Dati exemplu de o problema unde puteti defini (cel putin) doua euristici diferite si discutati care dintre ele este dominanta.
35. Dati exemplu de un algoritm studiat la curs care poate fi utilizat pentru determinarea ciclurilor de cost ngeativ intr-un graf orientat. Dupa aceea, scrieti pseudocodul pentru determinarea tuturor nodurilor care alcatuiesc un astfel de ciclu SAU demonstrati corectitudinea determinarii ciclului negativ de catre acest algoritm.
36. Algoritmul DFS poate fi folosit pentru clasificarea muchiilor unui graf orientat. Specificati tipurile de muchii si cum se poate determina tipul fiecarei muchii din graf folosind o singura parcurgere in adancime.
37. Care este inaltimea maxima la care poate ajunge un nod in cadrul algoritmilor de tip preflux (push-relabel) folositi pentru determinarea fluxului maxim. Argumentati pe scurt./ + De ce este utilă/necesară această limită?
38. Dandu-se matricea finala de distante obtinunta folosind algoritimul Floyd-Warshall, se poate stabili daca graful respectiv are un ciclu de cost negativ? Argumentati.
39. Demonstrati ca un graf are cicluri daca si numai daca in timpul unei parcurgeri DFS sunt descoperite arce noi.
40. Explicati utilitatea taieturilor alfa-beta pentru arborii de joc si dati un exemplu de arbore de joc in care sa evidentiati cel putin 2 taieturi:una pe nivelul MIN, a doua pe nivelul MAX./ Explicați ce reprezintă o tăietura alfa-beta într-un arbore de joc și dați un exemplu de astfel de arbore, cu maxim 3 nivele, în care să scoateți în evidență minim două tăieturi alfa-beta.
41. Pornind de la definitie, explicati de ce este important ca euristica folosita de algoritmul A\* sa fie admisibila? Demonstrati apoi ca orice euristica consistenta e admisibila.
42. Definiți ce înseamnă graful rezidual al unei rețele de transport și demonstrați (chiar și informal) că atunci când nu mai există nici o cale de ameliorare de la sursă la drenă în graful rezidual înseamnă că am descoperit tăietura minimală a rețelei de transport.
43. Dacă o problemă poate fi rezolvată corect atât prin greedy, cât și prin programare dinamică, ce metodă ați alege și de ce?
44. Pornind de la definiția funcției REVISE, arătați de ce algoritmul AC3 este mai perfomant decât algoritmul AC1.
45. Scrieți care este diferența între un flux și un preflux. Descrieți apoi pe scurt cum funcționează algoritmii de preflux pentru calculul fluxului maxim scoțând în evidența de ce la final prefluxul este un flux corect.
46. Ce se întâmplă cu algoritmul minimax cu tăieturi alfa-beta dacă modificăm condiția de tăiere din (β<α) în (β-k<α), unde k este o constantă oarecare. Discuție după k.
47. Construiti arborele Huffman de compresie pentru urmatorul text “ce abecedar” si calculati care este costul acestuia (cu 2 zecimale).
48. Dati un exemplu de graf cu maxim 5 noduri si cel putin o muchie de cost negativ pentru care algoritmul lui Dijkstra pentru drumuri minime de sursa unica calculeaza gresit costul drumurilor minime (este suficient sa scrieti pentru un nod costul corect si cel calculat de algoritm). Cum se poate modifica algoritmul pentru a rezolva aceasta problema?
49. Taieturile alfa-beta reduc complexitatea unei rezolvari cu minimax pentru jocuri doar daca sunt folosite impreuna cu o alta tehnica. Care este aceasta si de ce este importanta? Dati un scurt exemplu care demonstreaza ca folosirea acesteia impreuna cu alfa-beta aduce imbunatatiri.
50. Daca pentru fiecare nod n din cadrul unei explorari folosind algoritmul A\* euristica este definita in felul urmator: h(n) = max{cost(n, n’) | n’ ∈ succs(n)} (costul muchiei de cost maxim care pleaca din n catre un succesor al sau). Ce puteti spune despre aceasta euristica si despre rezultatul intors de catre algoritm?
51. Descrieti ce reprezinta o subproblema si determinati relatia de recurenta pentru problema parantezarii matricilor.
52. Dați exemplu de o problemă care se poate rezolva, eventual cu constrângeri diferite asupra datelor de intrare, atât folosind programare dinamică, cât și cu programare lacomă. Explicați diferențele și asemănările între cei doi algoritmi (inclusiv complexitatea lor).
53. Care este numărul minim, respectiv maxim, de componente tare conexe ale unui graf care admite o sortare topologică? Justificați!
54. Fie un algoritm A\* ghidat de următoarea euristică, h(n) = 0 ∀ n ∈ V, pentru problema 8- puzzle discutată la curs. Explicați care este numărul maxim de noduri care sunt scoase din mulțimea CLOSED pentru a fi reintroduse în mulțimea OPEN în timpul găsirii unei secvențe de pași de la o configurație inițială a problemei la o configurație finală. Asigură această euristică găsirea soluției optime ca număr de pași efectuați?