1. Mittedetermineeritud algoritmi tulemus samade lähteandmete korral võib erinevatel lahenduskordadel olla erinev. **Õige**

*Selgitus:* mittedetermineeritud käitumine võib sisaldada juhuslikkust.

1. Osaline algoritm peatub mistahes sisendandmete korral. **Vale**
2. Kahendotsimise keskmine ajaline keerukus on **Õige**
3. Ühildusmeetodi (*merge sort*) halvima juhu ajaline keerukus on **Vale**

*Selgitus:* See on

1. Järjestamismeetod on kiire, kui selle keskmine ajaline keerukus on **Õige**

*Selgitus:* Jah, üldjuhul ei saa kiiremini

1. Magasini (*stack*) viimati lisatud element eemaldatakse esimesena. **Õige**

*Selgitus:* LIFO = Last In First Out

1. Väärtusvaru on üks abstraktse andmetüübi komponent. **Õige**
2. Topeltseotud ahela iga element sisaldab viita nii eelmisele kui ka järgmisele elemendile. **Õige**
3. Avaldise pööratud poola kuju (RPN) saadakse parempoolsest suluesitusest sulgude ja komade ärajätmise teel. **Õige**
4. Iga täisgraaf on lihtgraaf. **Õige**
5. Iga nõrgalt sidus graaf on tugevalt sidus. **Vale**
6. Kui graafis esineb tsükkel, siis ei saa graafi tippe topoloogiliselt jarjestada. **Õige**
7. Rekursiooni saab magasini abil teisendada tsukliteks. **Õige**
8. Ammendava otsingu algoritmid on üldjuhul eksponentsiaalse ajalise keerukusega. **Õige**
9. Mida vaiksem on kahendotsimise puu kõrgus, seda efektiivsem on otsimine. **Õige**
10. Koodipuu abil saab kirjeldada prefikskoodi. **Õige**
11. Nullgraafi servade hulk on tühi. **Õige**
12. Lihtgraafis võivad esineda silmused. **Vale**
13. Kui graafis leidub tee tipust a tipuni b, siis selle graafi transitiivne sulund sisaldab kaart **Õige**
14. Toesepuu (*spanning tree*) on atsükliline. **Õige**
15. Funktsiooni keerukusklass on
16. Millist seost funktsioonide ja vahel valjendab järgmine definitsioon



Vali üks:

1. Millist seost funktsioonide ja vahel valjendab järgmine definitsioon



Vali üks:

1. Järjestamise ühildamismeetodi keskmine ajaline keerukus on
2. Järjestamise lihtsa pistemeetodi keskmine ajaline keerukus on
3. Järjestamise positsioonismeetodi keskmine ajaline keerukus on

1. Paisktabelist otsimise keskmine ajaline keerukus on
2. Kahendotsimise keskmine ajaline keerukus on
3. Järjestamise kuhjameetodi keskmine ajaline keerukus on
4. Järjestamise kiirmeetodi halvima juhu ajaline keerukus on

Vali üks:

1. Puu lehed on **alluvateta tipud**
2. Dijkstra algoritmiga arvutatakse graafis **antud tipust algavaid lühimaid teid kõigisse saavutatavatesse tippudesse**
3. Floyd-Warshalli algoritmiga arvutatakse graafis **lühimate teede pikkusi koigi tipupaaride vahel**
4. Kruskali algoritmiga arvutatakse graafis **minimaalset toesepuud**
5. Primi algoritmiga arvutatakse graafis **minimaalset toesepuud**
6. Milline algoritm kasutab tsükliliste räsifunktsioonide arvutamist alamsõne otsimiseks **Rabin-Karp**
7. Milline algoritm kasutab prefiksfunktsiooni arvutamist alamsõne otsimiseks **Knuth-Morris-Pratt**
8. Milline algoritm kasutab sufiksfunktsiooni ja ebasobiva sümboli heuristika arvutamist alamsõne otsimiseks

Vali üks:

**Boyer-Moore**

1. Milline algoritm kasutab sümbolihulkade poolitamist sümbolite koodide arvutamiseks **Shannon-Fano**
2. Milline algoritm kasutab koodipuu moodustamist sümbolite koodide arvutamiseks **Huffman**
3. Kui rekursiivne pöördumine on algoritmi viimane käsk, siis on see **sabarekursioon**
4. Tagasipöördumist varem korvale jäetud lahendusvariandi juurde ammendava otsingu ülesannetes nim. inglise keeles: ***backtracking***
5. Alamülesannete vastuste meeldejätmisel põhinevat iteratiivset lahendusmeetodit nim. inglise keeles: ***dynamic programming***
6. Kruskali algoritm (samuti Huffmani algoritm) on **ahne** **algoritm**
7. Pikima ühise osasõne leidmine (samuti Fibonacci jada moodustamine) on

*quicksort algorithm* / kiirsorteerimise algoritm

*exhaustive search algorithm* / ammendava otsingu algoritm

*greedy algorithm* / ahne algoritm

*divide and conquer algorithm* / jaga ja valitse algoritm

***dynamic programming algorithm* / dünaamilise kavandamise algoritm**

1. Lippude paigutamine malelauale, samuti seljakotiülesanne on ***exhaustive search algorithm* / ammendava otsingu algoritm**
2. Järjestamise kiirmeetod, samuti ühildamismeetod on ***divide and conquer algorithm* / jaga ja valitse algoritm**
3. Millise andmestruktuuriga on tegemist ***binary heap* / kahendkuhi**

****

1. Milline tippude järjestus saadakse läbides kahendpuud algoritmiga:

1) töödelda juur;

2) labida vasak alampuu;

3) labida parem alampuu.

***pre-order* / eesjärjestus**

1. Millist omadust kirjeldab lause: kõik võtmed vasakus alampuus ei ole suuremad juure võtmest ning kõik võtmed paremas alampuus ei ole väiksemad juure võtmest. ***property of keys in binary search tree* / kahendotsimise puu võtmete omadus**
2. Leia vastavus tähistuste ja tähenduste vahel, ja on funktsioonid, mille asümptootilist käitumist võrreldakse.

**kasvab kiiremini kui**

**kasvab aeglasemalt kui**

**kasvab mitte aeglasemalt kui**

**kasvab niisama kiiresti kui**

**kasvab mitte kiiremini kui**

1. Moodusta selle graafi lühimate teepikkuste maatriks täites lüngad arvudega (iga serva pikkus on üks ühik):

****

*Lahendus:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **A** | 0 | 1 | 3 | 1 | 2 |
| **B** | 1 | 0 | 2 | 1 | 1 |
| **C** | 3 | 2 | 0 | 2 | 1 |
| **D** | 1 | 1 | 2 | 0 | 1 |
| **E** | 2 | 1 | 1 | 1 | 0 |

1. Leia antud järjestuste hulgast kolm, mis sobivad selle graafi tippude topoloogiliseks järjestuseks.



BACFDEH

ADBCEHF

ACFDBEH

1. Leia allpool kolm järjestust, mis vastavad selle graafi tippude laiuti läbimise strateegiale alates tipust A.



ABDECF

ADBCEF

ABDCEF

1. Leia selle puu tippude lõppjärjestus ning esita see ilma tühikuteta sõnena.



Vastus: **GHDEIFBJCA**

1. Leia selle kahendpuu tippude keskjärjestus ning esita see ilma tühikuteta sõnena.



Vastus: **DBEAFC**

1. Leia selle avaldise pööratud poola kuju ning esita see sõnena, milles avaldise elemendid on eraldatud täpselt ühe tühikuga:

Vastus:

1. Progemine: Leia suurima laste arvuga tipu laste arv

Answer v =

Answer.parseTree("A(B,C(D,F(K,L,M,N(O)),P))");

System.out.println(v);

int n = v.maxWidth();

System.out.println("Maximum number of children: " +n); // 4

public class Answer {

private String name;

private Answer firstChild;

private Answer nextSibling;