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

Tõene

Väär

1. **Osaline algoritm (*Partial* *algorithm*) peatub mistahes sisendandmete korral.**

Tõene

Väär

1. **Kahendotsimise keskmine ajaline keerukus on O(log n).**

Tõene

Väär  Õige vastus on: Sisesta vastus siia

1. **Ühildusmeetodi (*merge* *sort*) halvima juhu ajaline keerukus on O(n).**

Tõene

Väär  O(n log n)

1. **Järjestamismeetod (*sorting* *method*) on kiire, kui selle keskmine ajaline keerukus on O(n log n).**

Tõene

Väär  Õige vastus on: Sisesta vastus siia

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

Tõene

Väär  Õige vastus on: Sisesta vastus siia

1. **Väärtusvaru on üks abstraktse andmetüübi komponent.**

Tõene

Väär

1. **Topeltseotud ahela iga element sisaldab viita nii eelmisele kui ka järgmisele elemendile.**

Tõene

Väär

1. **Avaldise pööratud poola kuju (RPN) saadakse parempoolsest suluesitusest sulgude ja komade ärajätmise teel.**

Tõene

Väär

1. **Iga täisgraaf on lihtgraaf.**

Tõene

Väär

1. **Iga nõrgalt sidus graaf on tugevalt sidus.**

Tõene

Väär

1. **Kui graafis esineb tsükkel, siis ei saa graafi tippe topoloogiliselt järjestada.**

Tõene

Väär

1. **Rekursiooni saab magasini abil teisendada tsükliteks.**

Tõene

Väär

1. **Ammendava otsingu algoritmid (*Exhaustive* *search* *algorithms*) on üldjuhul eksponentsiaalse ajalise keerukusega.**

Tõene

Väär

1. **Mida väiksem on kahendotsimise puu kõrgus, seda efektiivsem on otsimine.**

Tõene

Väär

1. **Koodipuu abil saab kirjeldada prefikskoodi.**

Tõene

Väär

1. **Nullgraafi servade hulk on tühi.**

Tõene

Väär

1. **Lihtgraafis võivad esineda silmused.**

Tõene

Väär

1. **Kui graafis leidub tee tipust a tipuni b, siis selle graafi transitiivne sulund sisaldab kaart (a,b).**

Tõene

Väär

1. **Toesepuu (*spanning* *tree*) on atsükliline.**

Tõene

Väär

1. **Funktsiooni keerukusklass on: 5n**
2. **Millist seost funktsioonide f ja g vahel väljendab järgmine definitsioon:**

****

Vali üks:

f ~ o(g)

f ~ ω(g)

f ~ Ω(g)

f ~ (g)

f ~ (g)

1. **Millist seost funktsioonide f ja g vahel väljendab järgmine definitsioon:**

****

Vali uks:

f ~ ω(g)

f ~ Ω(g)

f ~ (g)

f ~ o(g)

f ~ (g)

1. **Järjestamise ühildamismeetodi keskmine ajaline keerukus on: O(n log n)**
2. **Järjestamise lihtsa pistemeetodi keskmine ajaline keerukus on: O(n)**
3. **Järjestamise positsioonimeetodi keskmine ajaline keerukus on:**

O(n2)

O(nlogn)

O(n)

O(1)

O(log n)

1. **Paisktabelist otsimise keskmine ajaline keerukus on: O(1)**
2. **Kahendotsimise keskmine ajaline keerukus on:** O(log n)
3. **Järjestamise kuhjameetodi keskmine ajaline keerukus on: O(log n)**
4. **Järjestamise kiirmeetodi halvima juhu ajaline keerukus on:** O(n2)
5. **Puu lehed on:** alluvateta tipud
6. **Dijkstra algoritmiga arvutatakse graafis:** Antud tipust saavutatavaid lühimaid teid kõikidesse teistesse tipudesse
7. **Floyd-Warshalli algoritmiga arvutatakse graafis:** Kahe tipu vahelist lühimat teed
8. **Kruskali algoritmiga arvutatakse graafis:** minimaalset toesepuud
9. **Primi algoritmiga arvutatakse graafis:** minimaalset toesepuud
10. **Milline algoritm kasutab tsükliliste räsifunktsioonide arvutamist alamsõne otsimiseks: Rabin-Karp**
11. **Milline algoritm kasutab prefiksfunktsiooni arvutamist alamsõne otsimiseks:** Sisesta vastus siia
12. **Milline algoritm kasutab sufiksfunktsiooni ja ebasobiva sümboli heuristika arvutamist alamsõne otsimiseks:** Boyer-Moore
13. **Milline algoritm kasutab sümbolihulkade poolitamist sümbolite koodide arvutamiseks:** Morris-Pratt
14. **Milline algoritm kasutab koodipuu moodustamist sümbolite koodide arvutamiseks:** Huffmann
15. **Kui rekursiivne pöördumine on algoritmi viimane kask, siis on see Sabarekursioon**
16. **Tagasipöördumist varem kõrvale jäetud lahendusvariandi juurde ammendava otsingu ülesannetes nim. inglise keeles:** backtracking
17. **Alamülesannete vastuste meeldejätmisel põhinevat iteratiivset lahendusmeetodit nim. inglise keeles:** Dynamic Prograamming
18. **Kruskali algoritm (samuti Huffmani algoritm) on:** Ahne algoritm
19. **Pikima ühise osasõne leidmine (samuti Fibonacci jada moodustamine) on:**

kiirsorteerimise algoritm

ammendava otsingu algoritm

ahne algoritm

jaga ja valitse algoritm

dünaamilise kavandamise algoritm

1. **Lippude paigutamine malelauale, samuti seljakotiülesanne on** ammendava otsingu **algoritm.**
2. **Järjestamise kiirmeetod, samuti ühildamismeetod on** jaga ja valitse **algoritm/meetod.**
3. **Millise andmestruktuuriga on tegemist:**

****

Kahendkuhi

1. **Milline tippude järjestus saadakse läbides kahendpuud algoritmiga:**

1) töödelda juur;

2) läbida vasak alampuu;

3) läbida parem alampuu.

Pre-order ehk 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:** kahendpuu võtmete omadus
2. **Leia vastavus tähistuste ja tähenduste vahel, f ja g on funktsioonid, mille asümptootilist käitumist võrreldakse.**

f~ω(g) f kasvab kiiremini kui g

f~o(g) f kasvab aeglasemalt kui g

f~Ω(g) f kasvab mitte aeglasemalt kui g

f~(g) f kasvab niisama kiiresti kui g

f~(g) f kasvab mitte kiiremini kui g

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

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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.**



1. A C D E B H F
2. A B D E H C F
3. B A C D E H F
4. **Leia allpool kolm järjestust, mis vastavad selle graafi tippude laiuti läbimise strateegiale alates tipust A.**

****

1. A, D, E, B, C, F
2. A, D, B, E, F, C
3. A, B, C, F, E, D
4. **Leia selle puu tippude lõppjärjestus ning esita see ilma tühikuteta sõnena.**

GHDEBIFJCA

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

DBEFCA

1. **Leia selle avaldise pööratud poola kuju ning esita see sõnena, milles avaldise elemendid on eraldatud täpselt ühe tühikuga: 4 /(5-3) + 2\*6**

4 5 3 - / 2 6 \* +

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;