

I balle:

Paskaidrojiet un, ja nepieciešams, definējiet jautājumā uzdoto jēdzienu, terminu vai algoritmu. Atbildi papildiniet ar piemēru.

1. Datu, datu struktūras un datu tipa jēdziens.
2. Skalārais (vienkāršais, primitīvais) datu tips, atvasinātais datu tips, strukturētais datu tips.
3. Ordinālais datu tips. Sakārtotas un nesakārtotas datu struktūras.
4. Bāzes tips un indeksa tips, to lietojums.
5. Loģiskā datu struktūra un fiziskā datu struktūra.
6. Lineārā datu struktūra.
7. Nelineārā datu struktūra.
8. Biežāk lietotās datu struktūras, to raksturojums.
9. Deskriptors, tā lietojuma mērķis. Deskriptora uzbūve un tajā sakopotā informācija.
10. Masīva adresēšanas un ieraksta adresēšanas galvenās atšķirības.
11. Meklēšanas operācijas, izmantojot pozicionālo un asociatīvo piekļuvi.
12. Algoritmi elementa sameklēšanai vektorā.
13. Binārās meklēšanas algoritms un interpolatīvās meklēšanas algoritms, to salīdzināšana.
14. Steka un rindas pamatoperācijas, izmantojot hronoloģiski sakārtoto sarakstu.
15. Divkārssaisītais saraksts, tā izveidošanas un lietojuma specifika, priekšrocības un trūkumi.
16. Specifikācija, projektēšana un implementēšana (ieviešana).
17. Masīvs kā lineārā datu struktūra. Masīva specifikācija un pamatoperācijas.
18. Ieraksts kā lineārā datu struktūra. Ieraksta specifikācija un pamatoperācijas.
19. Ieraksts ar variantiem, tā specifikācija un lietojuma iespējas un noteikumi.
20. Hronoloģiski sakārtotais saraksts, tā izveidošanas un lietojuma pamatprincipi.
21. Pēc lietojuma biežuma sakārtots saraksts, tā izveidošanas un lietojuma pamatprincipi.
22. Pēc lietojuma biežuma sakārtotā sarakstu un pašorganizētā sarakstu salīdzinājums.
23. Trīs galvenās metodes pašorganizētā sarakstā izveidošanai un lietošanai.
24. Sašķirotais saraksts, tā izveidošanas un lietojuma pamatprincipi.
25. Datu struktūras pamatmodeļi pašorganizētā sarakstu veidošanai.
26. Datu struktūras pamatmodeļi sašķirotā sarakstu veidošanai.
27. Lineārās meklēšanas algoritma un binārās meklēšanas algoritma salīdzinājums.
28. Prioritātes rindas darbības princips, lietojuma modeļi, specifika un pamatoperācijas.
29. Steka darbības princips, steka modeļi un pamatoperācijas.
30. Rindas darbības princips, rindas modeļi un pamatoperācijas.
31. Steka attēlojums vektoriālajā formā un saistītajā formā, šo divu pamatmodeļu salīdzinājums.
32. Rindas attēlojums vektoriālajā formā un saistītajā formā, šo divu pamatmodeļu salīdzinājums.
33. Dekas darbības princips, attēlojuma modeļi un pamatoperācijas.
34. Tabulas un faila salīdzinājums pēc to uzbūves un elementu piekļuves.
35. Masīva un ieraksta salīdzinājums pēc to uzbūves un elementu piekļuves.
36. Rādītājs, tā specifikācija, tips, garums, un operācijas. Rādītāja lietojuma specifika un īpatnības.
37. Rādītāju masīvs un tā lietojums.
38. Vairākkārtsais saraksti, tā izveidošanas pamatprincipi.
39. Ierakstu masīvs un masīvu ieraksts, to izveidošana un piekļuve.
40. Strukturētie datu tipi un to lietojums sarežģītu datu struktūru veidošanai.
41. Vektoriālajā formā un saistītajā formā attēloto sarakstu salīdzinājums.

42. Vienkāršsaistīto un divkāršsaistīto sarakstu salīdzinājums.
43. Datu struktūru modeļi saistīto sarakstu veidošanai.
44. Vienkāršsaistītājā un divkāršsaistītājā cirkulārā saraksta veidošanas pamatprincipi un modeļi.
45. Vektoriālajā formā attēlotais steks, tā trūkumi un to novēršana.
46. Vektoriālajā formā attēlotā rinda, tās trūkumi un to novēršana.
47. Cirkulārā rinda, tās izveidošanas un lietojuma specifika.
48. Rakstzīmju virknes tips String un tā izveidošanas paņēmieni.

II balles:

1. Izveidot deskriptoru un adresēšanas funkciju AMF masīvam:
 - a) var Arr1: array[0 .. 5, -10 .. 10] of real;
 - b) var Arr2: array[-2 .. 2, 0 .. 4] of integer;
 - c) var Arr3: array[1 .. 3, 0 .. 2, -1 .. 1] of char;
 ja bāzes adrese b=1000.
2. Paskaidrot, kādu metodiku lieto, lai noteiktu vairākdimensiju masīva adresēšanas funkciju AMF. Noteikt adresēšanas funkciju AMF masīvam:
 - a) var A: array [-1 .. 1, 0 .. 2, 1 .. 5, -1 .. 1] of integer;
 - b) var B: array [0 .. 2, -1 .. 1, 0 .. 1, 1 .. 5] of real;
 - c) var C: array [-1 .. 2, 0 .. 1, -1 .. 1, 0 .. 4] of char;
 ja bāzes adrese b=500.
3. Dot piemēru ierakstam ar variantiem. Izveidot tā aprakstu. Paskaidrot, kā ieraksta lauki apstrādei pieejami.
4. Izveidot ieraksta nobīžu sarakstu:
 - a) type R = record
 - a: 1 .. 100;
 - b: real;
 - c: array [1 .. 10] of char;
 - d: boolean;
 end;
 var Z: R;
 - b) type Rec = record
 - x: string[10];
 - y: record
 - dy: 1 .. 31;
 - mo: 1 .. 12;
 - yr: 1900 .. 2100
 end
 var Q: Rec;
5. Dot piemēru ierakstam, kurā ir vismaz 5 lauki. Izveidot šī ieraksta aprakstu un nobīžu sarakstu. Paskaidrot, kādam nolūkam nobīžu saraksts paredzēts.

6. Dot

- a) diagonālmaticas;
- b) simetriskās matricas;
- c) augšējās trīsstūrmaticas;
- d) apakšējās trīsstūrmaticas;
- e) retinātās matricas

piemēru.

Aprakstīt tās attēlojumu vektoriālajā formā un izveidot:

- a) operāciju matricas elementu ievadei un operāciju elementa sameklēšanai vektorā;
- b) operāciju matricas elementu izvadei un operāciju elementa sameklēšanai vektorā.

7. Dot

- a) ierakstu masīva (tabulas);
- b) masīvu ieraksta;
- c) rādītāju masīva

piemēru.

Aprakstīt šo datu struktūru. Paskaidrot šīs datu struktūras elementu piekļuvi un lietojuma specifiku.

8. Ar piemēru ilustrēt retinātās matricas divus attēlojuma paņēmienus.

9. Izvēlēties virknes rakstzīmju lietojuma biežuma tabulu, kurā ir vismaz 6 rakstzīmes, un izveidot Hafmena kodu.

10. Izveidot vektoriālajā formā attēlotā saraksta modeli, aprakstīt to un izveidot operāciju FindKey lineārās meklēšanas, robežmarķiera metodes un binārās meklēšanas algoritmiem elementa sameklēšanai vektorā..

11. Izveidot saistītajā formā attēlotās rakstzīmju virknes modeli, aprakstīt to un izveidot operāciju:

- a) Concatenate;
- b) Substring;
- c) Insert;
- d) Delete;
- e) ReadString;
- f) WriteString.

12. Izveidot vektoriālā formā attēlotās rakstzīmju virknes modeli, aprakstīt to un izveidot operāciju:

- a) Fill - kas virkni, sākot ar pozīciju pos līdz pozīcijai len, aizpilda ar rakstzīmi c;
- b) StrCopy - kas len rakstzīmes no virknes S1 kopē virknē S2. Ja virknes tekošais garums ir mazāks par len, tad kopē tik rakstzīmju, cik to ir virknē S1;
- c) Distribute - kas virkni S1, sākot no pozīcijas pos, sadala 2 daļās - virknēs S1 un S2;
- d) EscSpace - kas likvidē visas tukšumzīmes virknes sākumā un beigās;
- e) OneBlank - kas likvidē tukšumzīmes virknes tekstā starp rakstzīmēm, ja šādu tukšumzīmju skaits ir lielāks par vienu.

13. Izveidot vektoriālajā formā attēlotā saraksta modeli, aprakstīt to un izveidot 2 operācijas FindKey

atbilstoši binārās meklēšanas un interpolatīvās meklēšanas algoritmam.

14. Izveidot modeli vienkāršsaistītajam sarakstam:

- a) bez beigu rādītāja tail;
- b) ar beigu rādītāju tail;
- c) cirkulārajam sarakstam,

aprakstīt to un izveidot operāciju **Insert**, kas jaunu elementu pievieno:

- a) saraksta sākumā;
- b) saraksta beigās;
- c) aiz i-tā elementa;
- d) pirms i-tā elementa.

15. Izveidot modeli divkāršsaistītajam sarakstam:

- a) bez beigu rādītāja tail;
- b) ar beigu rādītāju tail;
- c) cirkulārajam sarakstam,

aprakstīt to un izveidot operāciju **Insert**, kas jaunu elementu pievieno:

- a) saraksta sākumā;
- b) saraksta beigās;
- c) aiz i-tā elementa;
- d) pirms i-tā elementa.

16. Izveidot modeli vektoriālajā formā attēlotajam sarakstam, aprakstīt to un izveidot operāciju **Insert** kas jaunu elementu pievieno:

- a) saraksta sākumā;
- b) saraksta beigās;
- c) aiz i-tā elementa;
- d) pirms i-tā elementa.

17. Izveidot modeli vienkāršsaistītajam sarakstam:

- a) bez beigu rādītāja tail;
- b) ar beigu rādītāju tail;
- c) cirkulārajam sarakstam,

aprakstīt to un izveidot operāciju **Delete** tā, lai par tekošo elementu kļūst:

- a) tekošā elementa priekštecis;
- b) tekošā elementa pēctecis;
- c) pirmais elements;
- d) pēdējais elements.

18. Izveidot modeli divkāršsaistītajam sarakstam:

- a) bez beigu rādītāja tail;
- b) ar beigu rādītāju tail;
- c) cirkulārajam sarakstam,

aprakstīt to un izveidot operāciju **Delete** tā, lai par tekošo elementu kļūst:

- a) tekošā elementa priekštecis;
- b) tekošā elementa pēctecis.
- c) pirmais elements;
- d) pēdējais elements.

19. Izveidot modeli:

- a) vienkāršsaistītajam sarakstam;
- b) divkāršsaistītajam sarakstam,

aprakstīt to un izveidot operāciju **Swap**, kas tekošo elementu apmaina vietām ar:

- a) iepriekšējo elementu;
- b) nākamo elementu;
- c) pirmo elementu;
- d) pēdējo elementu;
- e) i-to elementu.

20. Izveidot modeli vektoriālajā formā attēlotajam sarakstam, aprakstīt to un izveidot operāciju **Delete** tā, lai par tekošo elementu kļūst:

- a) tekošā elementa priekštecis;
- b) tekošā elementa pēctecis.
- c) pirmais elements;
- d) pēdējais elements.

21. Izveidot modeli:

- a) vienkāršsaistītajam necirkulārajam sarakstam;
- b) divkāršsaistītajam necirkulārajam sarakstam;
- c) vienkāršsaistītajam cirkulārajam sarakstam;
- d) divkāršsaistītajam cirkulārajam sarakstam;
- e) vektoriālajā formā attēlotajam sarakstam,

aprakstīt to un izveidot operāciju **PrintList** visu saraksta elementu izvadei ekrānā.

III balles:

1. Izveidot modeli hronoloģiski sakārtotajam sarakstam, kas attēlots:

- a) vektoriālajā formā;
- b) kā vienkāršsaistītas necirkulārais saraksts;
- c) kā divkāršsaistītais necirkulārais saraksts;
- d) kā vienkāršsaistītais cirkulārais saraksts;
- e) kā divkāršsaistītais cirkulārais saraksts,

aprakstīt to un izveidot operācijas:

- a) FindKey, Insert, Retrieve;
- b) Findith, Delete, Update;
- c) FindFirst, FindLast, Insert, PrintList.

2. Izveidot modeli pašorganizētajam sarakstam, kas veidots pēc:

- a) 1.metodes;
- b) 2.metodes;
- c) 3.metodes

un attēlots:

- a) vektoriālajā formā;
- b) kā vienkāršsaistītais necirkulārais saraksts;
- c) kā divkāršsaistītais necirkulārais saraksts;
- d) kā vienkāršsaistītais cirkulārais saraksts;
- e) kā divkāršsaistītais cirkulārais saraksts.

Izveidot operācijas:

- a) Findith, Insert, Update;
- b) FindKey, Delete, Retrieve;
- c) FindLast, FindKey, Insert, FindNext.

3. Izveidot modeli gredzenam, kas attēlots:

- a) vektoriālajā formā;
- b) kā vienkāršsaistītais saraksts;
- c) kā divkāršsaistītais saraksts,

aprakstīt to un izveidot operācijas:

- a) InsertBefore, FindKey, Update;
- b) InsertAfter, FindPrior, Retrieve;
- c) PrintRing, Delete (par tekošo kļūst iepriekšējais elements), InsertAfter;
- d) FindNext, Delete (par tekošo kļūst nākamais elements), FindKey.

4. Izveidot modeli vienkāršsaistītajam cirkulāram sarakstam, aprakstīt to un izveidot operācijas:

- a) Findith, InsertBefore, Update;
- b) FindKey, InsertAfter, Terminate;
- c) FindFirst, Delete (par tekošo kļūst iepriekšējais elements), PrintList;
- d) FindLast, Delete (par tekošo kļūst nākamais elements), FindPrior.

5. Izveidot modeli divkāršsaistītajam cirkulāram sarakstam, aprakstīt to un izveidot operācijas:

- a) Findith (uzlabotais algoritms), InsertBefore, Delete (par tekošo kļūst nākamais elements);
- b) FindKey, InsertAfter, Update;
- c) FindLast, Terminate, Delete (par tekošo kļūst iepriekšējais elements);
- d) InsertBefore (kas neizsauc InsertAfter), Findkey, FindFirst.

6. Izveidot modeli vektoriālajā formā attēlotajam sašķīrotam sarakstam, aprakstīt to un izveidot operācijas:

Sort - kas saraksta elementus sašķiro augošā secībā pēc atslēgas lauka key vērtībām;
Findkey1 - kas realizē lineārās meklēšanas algoritmu;
Findkey2 - kas realizē binārās meklēšanas algoritmu.

7. Izveidot modeli sašķīrotajam sarakstam, kas attēlots:

- a) vektoriālajā formā;
- b) kā vienkāršsaistītais saraksts bez beigu rādītāja tail;
- c) kā vienkāršsaistītais saraksts ar beigu rādītāja tail;
- d) kā vienkāršsaistītais cirkulārais saraksts;
- e) kā divkāršsaistītais saraksts bez beigu rādītāja tail;
- f) kā divkāršsaistītais saraksts ar beigu rādītāja tail;
- g) kā divkāršsaistītais cirkulārais saraksts,

aprakstīt to un izveidot operācijas:

- a) FindKey, Delete (par tekošo kļūst iepriekšējais elements), Terminate;
- b) Update, Findith, Delete (par tekošo kļūst nākamais elements);
- c) Insert, FindKey, FindFirst,.
- d) Findith, Update, FindPrior, FindLast.

8. Izveidot modeli stekam, kas attēlots:

- a) vektoriālajā formā;
- b) saistītajā formā;
- c) kā steka pāris,

aprakstīt to un izveidot 3 operācijas:

- a) Retrieve - kas nolasa i -to elementu, to nedzēšot;
Cancel - kas dzēš to elementu, kura atslēgas lauka vērtība ir tkey;
Swap - kas steka virsotnē esošo elementu apmaina vietām ar to elementu, kas atrodas pozīcijā i .
- b) Retrieve - kas nolasa to elementu, kura atslēgas lauka vērtība ir tkey, elementu nedzēšot;
Cancel - kas dzēš i -to elementu;
Top - kas uz steka virsotni pārceļ to elementu, kura atslēgas lauka vērtība ir tkey.

9. Izveidot modeli rindai, kas attēlotā:

- a) saistītajā formā;
- b) kā cirkulārā rinda;
- c) vektoriālajā formā,

aprakstīt to un izveidot operācijas:

- a) Retrieve - kas nolasa to elementu, kura atslēgas lauka vērtība ir tkey, to nedzēšot;
Cancel - kas dzēš i -to elementu;
Top - kas uz rindas sākumu pārceļ to elementu, kura atslēgas lauka vērtība ir tkey.
- b) Retrieve - kas nolasa i -to elementu, to nedzēšot;
Cancel - kas dzēš to elementu, kura atslēgas laika vērtība ir tkey;
Swap - kas rindas sākumā esošo elementu apmaina vietām ar to elementu, kas atrodas pozīcijā i .

10. Izveidot modeli prioritātes rindai, kas attēlots:

- a) izmantojot vektoriālo modeli;
- b) izmantojot saistīto necirkulāru modeli;
- c) izmantojot saistītu cirkulāro modeli,

aprakstīt to un izveidot operācijas Enqueue, Serve, Size.

11. Izveidot modeli dekam, kas attēlots:

- a) izmantojot vektoriālo modeli;
- b) izmantojot saistītu necirkulāru modeli;
- c) izmantojot saistītu cirkulāro modeli,

aprakstīt to un izveidot operācijas Enqueue, Serve, Size.

12. Izveidot saistītajā formā attēlotās rakstzīmju virknes modeli, aprakstīt to un izveidot operācijas:

- a) Concatenate, Delete, MakeEmpty;
- b) Insert, Substring, Append;
- c) ReadString, Substring, Remove.