

Mustaqil ishlash uchun masalalar

1-masala: Tanlovni saralash (Selection sort) algoritmidan foydalanib elementlar ro'yxatini kamayish tartibida saralash uchun dasturi yozing. (Tanlovni saralash oddiy va samarali tartiblash algoritmi bo'lib, ro'yxatning tartiblanmagan qismidan eng kichik (yoki eng katta) elementni qayta-qayta tanlab, uni ro'yxatning tartiblangan qismiga ko'chirish orqali ishlaydi.)

Dastur natijasi :

```
Massiv elementlari sonini kiriting : 10
Massiv elementlarini kiriting: 7 8 9 2 4 5 6 3 4 7 8
Saralangan massiv : 9 8 7 7 6 5 4 4 3 2

Process returned 0 (0x0)   execution time : 11.830 s
Press any key to continue.
```

2-masala: Pufakchali tartiblash (Bubble sort) algoritmidan foydalanib elementlar ro'yxatini tartiblash uchun dastur yozing. (Bubble Sort - bu eng oddiy tartiblash algoritmi bo'lib, agar ular noto'g'ri tartibda bo'lsa, ulashgan elementlarni qayta-qayta almashtirish orqali ishlaydi. Ushbu algoritm katta ma'lumotlar to'plamlari uchun mos emas, chunki uning o'rtacha va eng yomon vaqt sarflashi ancha yuqori.)

Dastur natijasi :

```
Berilgan massiv : 15 56 12 -21 1 659 3 83 51 3 135 0
Tartiblangan massiv : -21 0 1 3 3 12 15 51 56 83 135 659

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```

3-masala: Tartiblash (insertion sort algorithm.) algoritmidan foydalanib elementlar ro'yxatini tartiblash uchun dastur yozing. (Insertion sort algorithm - bu oddiy tartiblash algoritmi bo'lib, u sizning qo'lingizda o'yin kartalarini saralash usuliga o'xshash ishlaydi. Massiv deyarli tartiblangan va tartiblanmagan qismga bo'lingan. Saralanmagan qismdan qiymatlar tanlanadi va tartiblangan qismning to'g'ri joyiga joylashtiriladi.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
10 ta elementlarini kiriting : 4 5 8 3 12 65 41 21 87 52 33
tartiblangan massiv : 3 4 5 8 12 21 41 52 65 87

Process returned 0 (0x0)   execution time : 17.603 s
Press any key to continue.
```

4-masala: Birlashtirish tartiblash (merge sort) algoritmidan foydalangan holda massiv elementlar ro'yxatini saralash uchun dastur yozing.(Birlashtirish saralash (merge sort) massivni kichikroq kichik massivlarga bo'lish, har bir kichik massivni saralash va so'ng saralangan pastki massivlarni yana birlashtirib, yakuniy tartiblangan massivni hosil qilish orqali ishlaydigan tartiblash algoritmi sifatida aniqlanadi.)

Dastur natijasi :

```
Berilgan massiv : 125 181 130 25 61 887 45 21 32 75
Tariblangan massiv : 21 25 32 45 61 75 125 130 181 887

Process returned 0 (0x0)   execution time : 0.030 s
Press any key to continue.
```

5-masala: MAX heap algoritmidan foydalanib raqamlarni tartiblash uchun dastur yozing.(Max-Heap har bir ichki tugun (node) o'z vorislariga teng yoki kattaroq bo'lgan yig'ma ma'lumotlar strukturasi turi sifatida aniqlanadi.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
Massiv elementlarini kiriting : 4 2 8 32 12 41 54 65 32 10
Tariblangan massiv : 2 4 8 10 12 32 32 41 54 65

Process returned 0 (0x0)   execution time : 13.350 s
Press any key to continue.
```

6-masala: Tez saralash (quick sort) algoritmidan foydalanib elementlar ro'yxatini saralash uchun dastur yozing.(QuickSort – bu “Bo'l va zabt et” algoritmiga asoslangan saralash algoritmi bo'lib, u elementni aylanma sifatida tanlaydi va berilgan massivni saralangan massivda to'g'ri joyiga qo'yib, berilgan massivni tanlangan pivot atrofida ajratadi.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
Massiv elementlarini kiriting : 2
6
45
21
32
98
75
54
65
23
Tez saralashgandan so'ng massiv : 2 6 21 23 32 45 54 65 75 98
Process returned 0 (0x0)   execution time : 11.399 s
Press any key to continue.
```

7-masala: Cocktail Sort usuli yordamida raqamlarni tartiblovchi dasturyozing.(Cocktail Sort - Bubble sortining o'zgarishi. Bubble tartiblash algoritmi har doim elementlarni chapdan kesib o'tadi va eng katta elementni birinchi iteratsiyada to'g'ri joyiga, ikkinchi iteratsiyada esa ikkinchi eng katta elementga o'tkazadi va hokazo. Kokteyl Saralash berilgan massiv bo'ylab har ikki yo'nalishda ham o'tadi. Kokteyllarni saralash keraksiz iteratsiyadan o'tmaydi, bu esa uni katta massivlar uchun samarali qiladi.)

Dastur natijasi :

```
-199 -52 2 3 33 56 99 100 177 200
Process returned 0 (0x0)   execution time : 0.038 s
Press any key to continue.
```

8-masala: Ikkilik kiritish (Binary insertion sort) yordamida raqamlarni tartiblovchi dastur yozing.(Ikkilik kiritish tartiblash qo'shish tartibiga o'xshash tartiblash algoritmidir, lekin element qo'shilishi kerak bo'lgan joyni topish uchun chiziqli qidiruvdan foydalanish o'rniga biz ikkilik qidiruvdan foydalanamiz. Shunday qilib, biz $O(N)$ dan $O(\log N)$ ga bitta elementni kiritishning qiyosiy qiymatini kamaytiramiz.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
Massiv elementlarini kiriting :
5
3
2
1
8
4
6
12
32
44
Tartiblangan massiv :
1 2 3 4 5 6 8 12 32 44
Process returned 0 (0x0)   execution time : 11.648 s
Press any key to continue.
```

Mavzu yuzasidan testlar:

1. Qidiruv va saralash algoritmlari qaysi ma'lumotlar tuzilishi bilan eng yaxshi amalga oshiriladi?
 - a) Massivga asoslangan ro'yxat
 - b) Bog'langan ro'yxat
 - c) Yuqoridagi ikkalasi ham*
 - d) Yuqoridagilarning hech biri
2. Qidiruv algoritmda ishlatiladigan kalit nima?
 - a) Qidirish, saralash, kiritish va o'chirish kabi operatsiyalarda ishlatiladi*
 - b) Ma'lumotlarni shifrlash uchun ishlatiladi
 - c) CompareTo() va equals() operatsiyalarida ishlatiladi
 - d) Yuqoridagilarning hech biri
3. Qaysi saralash algoritmi har bir o'tish uchun bitta almashtirishni amalga oshiradi (ehtimol elementni o'zi bilan almashtirish mumkin)?
 - a) Tanlab Saralash (Selection Sort)*
 - b) Pufakchali Saralash (Bubble Sort)
 - c) Qo'shish Tartibi (Insertion Sort)
 - d) Tog'ri javob berilmagan
4. Qaysi saralash algoritmi har bir o'tish uchun bir nechta almashtirishni amalga oshirishi mumkin?

- a) Tanlab Saralash (Selection Sort)
- b) Pufakchali Saralash (Bubble Sort)*
- c) Qo'shish Tartibi (Insertion Sort)
- d) Tog'ri javob berilmagan

5. Qaysi saralash algoritmi har bir o'tishda kerak bo'lsa elementlarni almashtirish o'rniga elementlarni o'zgartiradi.

- a) Tanlab Saralash (Selection Sort)
- b) Pufakchali Saralash (Bubble Sort)
- c) Qo'shish Tartibi (Insertion Sort)*
- d) Tog'ri javob berilmagan

6. Quyidagi ro'yxatlar saralash algoritmining 3 ta o'tishini ifodalaydi. Ro'yxatni saralash uchun qaysi algoritmdan foydalanilmoqda?

4 5 9 6 2 7

4 5 6 2 7 9

4 5 2 6 7 9

- a) Tanlab Saralash (Selection Sort)
- b) Pufakchali Saralash (Bubble Sort)*
- c) Qo'shish Tartibi (Insertion Sort)
- d) Tog'ri javob berilmagan

7. Biz quyidagi ro'yxatni o'sish tartibida saralaymiz:

1 4 2 9 3 8 5

Bubble sort algoritmidan bir marta o'tgandan keyin ro'yxat nimaga o'xshaydi.

- a) 1 2 4 3 8 5 9*
- b) 1 4 2 5 3 8 9
- c) 4 2 9 3 8 5 1
- d) 1 4 2 9 3 8 5

8. "Bo'l va zabt et" so'zi quyidagilarga ishora qiladi:

- a) Ro'yxat har birida teng miqdordagi elementlarga ega bo'lgan pastki ro'yxatlarga bo'linadi, so'ngra tartiblangan pastki ro'yxatlar yana birlashtiriladi.
- b) Ro'yxat kichikroq pastki ro'yxatlarga bo'linadi, so'ngra tartiblangan pastki ro'yxatlar yana birlashtiriladi.*
- c) Ro'yxat faqat ikkita pastki ro'yxatga bo'lingan, hech qachon ko'p yoki kamroq, ular saralanadi va birlashtiriladi.
- d) Tog'ri javob berilmagan

9. Tez saralash algoritmining eng yomon vaqt murakkabligi qaysi?

- a) $O(n^2)$ *
- b) $O(\log n)$
- c) $O(1)$
- d) $O(\log_2 n)$

10. Tez saralash (quick sort) algoritmi butun massivni nechta kichik massivga ajratadi?

- a) Bitta
- b) Ikkita*
- c) Uchta
- d) to'rtta