

3 - AMALIY MASHG'ULOT. SARALASH ALGORITMLARI. ODDIY ALGORITMLAR.

Ishdan maqsad: Talabalarni saralash va oddiy algoritmlar bilan tanishtirish. C++ dasturlash tilida algoritmlarga oid misollar yaratish.

Nazariy qism. Saralash algoritmlari informatika sohasidagi asosiy element hisoblanadi. Ular ko'plab muammolarni hal qilish uchun zarur bo'lgan ma'lumotlar to'plamini ma'lum bir tartibda tartibga solish imkonini beradi. Muayyan algoritmlarni ko'rib chiqishdan oldin, ularning ishlashini baholaydigan ishlash mezonlarini tushunish muhimdir. Vaqtning murakkabligi, makonning murakkabligi, barqarorligi, ko'p qirraliligi va boshqa mezonlar bizga ma'lum bir muammo uchun eng mos algoritmnı tanlashga yordam beradi.

Eng oddiy tartiblash algoritmlari

Pufakchali tartiblash bo'yicha Saralash

Bubble sort eng oddiy va intuitiv tartiblash algoritmlaridan biridir. U ro'yxatni bir necha marta ko'rib chiqadi, har bir qo'shni elementlar juftligini taqqoslaydi va agar ular noto'g'ri tartibda bo'lsa, ularni almashtiradi. Ro'yxat to'liq tartiblanmaguncha bu jarayonni takrorlaydi.

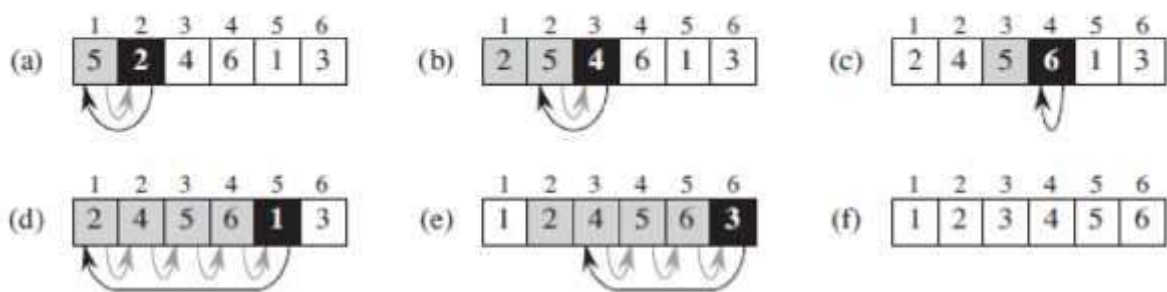
Tanlash bo'yicha Saralash

Tanlovni saralash quyidagicha ishlaydi: har bir qadamda algoritm qolgan elementlardan eng kichik elementni tanlaydi va uni birinchi element bilan almashtiradi. Keyin ro'yxatning qolgan qismi uchun bu jarayonni davom ettiradi.

Kiritish tartibi bo'yicha Saralash

Qo'shishni tartiblash dastlabki ikkita elementni saralashdan boshlanadi va keyin qolgan elementlarni ro'yxatning tartiblangan qismiga ketma-ket kiritadi.

Qo'shish tartibi - bu oz sonli elementlarni saralash uchun samarali algoritm.



Massivda INSERT-SORT amali. Massiv indeksleri to'rtburchaklar ustida ko'rsatiladi va massiv pozitsiyalarida saqlangan qiymatlar to'rtburchaklar ichida ko'rsatiladi. (a)–(e) 1–8-qatorlardagi for tsiklining takrorlanishi. Har bir iteratsiyada qora to'rtburchakda dan olingan

kalit mavjud bo'lib, u 5-qator testida uning chap tomonidagi soyalı qutilardagi qiymatlar bilan taqqoslanadi. Soyali strelkalar massiv qiymatlarining bir pozitsiyaga ko'chirilganligini ko'rsatadi. 6-qatorning o'ng tomonida va qora o'qlar 8-qatorda kalitning qayerga ko'chirilishini ko'rsatadi. (f) - yakuniy tartiblangan massiv. $A = \langle 5, 2, 4, 6, 1 \rangle A[j]$

Amaliy qism.

1-misol: Butun sonlar massivining pufakcha turi.

```
#include <iostream>
void bubbleSort (int arr [], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < ni-1; j++ ) {
            if ( arr [j] > arr [j+1]) {
                int temp = arr [j];
                arr [j] = arr [j+1];
                arr [j+1] = temp;
            }
        }
    }
}
int main() {
    int arr [] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof ( arr ) / sizeof ( arr [0]);
    bubbleSort ( arr , n );
    std:: cout << "Tartiblangan massiv:";
    for (int i = 0; i < n; i++)
        std:: cout << " " << arr [ i ];
    std:: cout << std:: endl ;
    return 0;
}
```

2-misol: tanlash usuli yordamida qatorlar massivini saralash.

```
#include <iostream>
#include <algorithm>
#include <string>
void selectionSort (std::string arr [], int n) {
    (int i = 0; i < n-1; i++) {
        int min_idx = i;
        for (int j = i+1; j < n; j++ ) {
            if ( arr [j] < arr [ min_idx ])
                min_idx = j;
        }
    }
}
```

```

std::swap( arr [ i ], arr [ min_idx ] );
    }
}
int main() {
std::string arr [] = {"olma", "banan", "gilos", "sana", "anjir"};
int n = sizeof ( arr ) / sizeof ( arr [0]);
    selectionSort ( arr , n);
std:: cout << "Tartiblangan massiv:";
for (int i = 0; i < n; i++)
std:: cout << " " << arr [ i ];
std:: cout << std:: endl ;
    return 0;
}

```

1-topshiriq: Massivdagi k-eng kichik elementni topish.

```

#include <iostream>
#include <algorithm>
int kthSmallest (int arr [], int n, int k) {
std::sort( arr , arr + n);
return arr [k-1];
}
int main() {
int arr [] = {7, 10, 4, 3, 20, 15};
int n = sizeof ( arr ) / sizeof ( arr [0]);
int k = 3;
int result = kthSmallest ( arr , n, k);
std:: cout << "" << k << "- eng kichik element: " << result << std:: endl;
    return 0;
}

```

2-topshiriq: massivni buyurtma uchun tekshirish.

```

#include <iostream>
bool isSorted (int arr [], int n) {
for (int i = 1; i < n; i++) {
if ( arr [ i ] < arr [i-1])
return false;
}
Return true;
}
int main() {
int arr [] = {1, 3, 5, 7, 9};
int n = sizeof ( arr ) / sizeof ( arr [0]);

```

```

std::cout << "Masiv tartiblanganmi? " << ( isSorted ( arr , n) ? "Ha" : "Yo'q") <<
std::endl ;
return 0;
}

```

3-misol: O'rnatilgan funksiya yordamida massivni saralash.

```

#include <iostream>
#include <algorithm>
int main() {
int arr [] = {3, 1, 4, 1, 5, 9, 2, 6, 5};
int n = sizeof ( arr ) / sizeof ( arr [0]);
std::sort( arr , arr + n);
std::cout << "Tartiblangan massiv:";
for (int i = 0; i < n; i++)
std::cout << " " << arr [ i ];
std::cout << std::endl ;
return 0;
}

```

3-topshiriq: massivni katta-kichik harflarga qarab tartiblash.

```

#include <iostream>
#include <algorithm>
#include <string>
int main() {
std::string arr [] = {"Olma", "banan", "Gilos", "sana", "anjir"};
int n = sizeof ( arr ) / sizeof ( arr [0]);
std::sort( arr , arr + n, [](const std::string& a, const std::string& b) {
return std::tolower (a) < std::tolower (b);
});
std::cout << "Tartiblangan massiv:";
for (int i = 0; i < n; i++)
std::cout << " " << arr [ i ];
std::cout << std::endl ;
return 0;
}

```

4-misol: ma'lum bir kalit bo'yicha kortejlar massivini tartiblang.

```

#include <iostream>
#include <algorithm>
#include <tuple>
int main() {
std::tuple<std::string, int> arr [] = {{"olma", 2}, {"banan", 1}, {"gilos", 3}};
int n = sizeof ( arr ) / sizeof ( arr [0]);

```

```

std::sort( arr , arr + n, [](const std::tuple<std::string, int>& a, const
std::tuple<std::string, int>& b) {
return std::get<1>(a) < std::get<1>(b);
});
std::cout << "Tartiblangan massiv:";
for (int i = 0; i < n; i++)
std::cout << " (" << std::get<0>( arr [ i ]) << " , " << std::get<1>( arr [ i ]) <<
")";
std::cout << std::endl ;
return 0;
}

```

4-topshiriq: Muayyan maydon bo'yicha tuzilmalar massivini saralash.

```

#include <iostream>
#include <algorithm>
struct Person {
std::string name;
int age;
};
bool compareByAge (const Person&a, const Person& b) {
return a.age < b.age;
}
int main() {
Person arr [] = {{"Jon", 30}, {"Alis", 25}, {"Bob", 35}};
int n = sizeof ( arr ) / sizeof ( arr [0]);
std::sort( arr , arr + n, compareByAge );
std::cout << "Yosh bo'yicha saralangan massiv:";
for (int i = 0; i < n; i++)

```

Mustaqil bajarish uchun topshiriqlar

Saralash algoritmi bo'yicha:

1. Pufakcha usuli yordamida butun sonlar massivini saralash.
2. Tanlash usuli yordamida qatorlar ro'yxatini saralash.
3. Qo'shish usuli yordamida haqiqiy sonlar massivini saralash.
4. Sanalar ro'yxatini o'sish tartibida tartiblang.
5. Pufak usuli yordamida mantiqiy qiymatlar massivini saralash.
6. Talabalar ro'yxatini GPA bo'yicha tartiblang.
7. Belgilar massivini alifbo tartibida tartiblash.
8. Maxsus sinf ob'ektlari ro'yxatini ularning xususiyatlaridan biri bo'yicha saralash.

9. Juft sonlar massivini yig'indisi bo'yicha saralash.

10. Ismlar ro'yxatini uzunligi bo'yicha saralash.

Eng oddiy algoritmlar bo'yicha:

1. Butun sonlar massivi uchun pufakchali tartiblash algoritmini amalga oshiruvchi C++ dasturini yozing.

2. C++ tilida satrlar ro'yxatini tanlash yordamida tartiblovchi funksiya yozing.

3. C++ tilida haqiqiy sonlar massivi uchun qo'shishni tartiblash algoritmini amalga oshiring.

4. Sanalar massivini o'sish tartibida tartiblovchi C++ dasturini yozing.

5. Mantiqiy qiymatlar massivi uchun C++ da pufakchali tartiblashni amalga oshiring.

6. C++ tilida talabalar ro'yxatini GPA bo'yicha tartiblovchi funksiyani yozing.

7. C++ tilida alfavit tartibida belgilar massivini tartiblashni amalga oshirish.

8. Maxsus sinf ob'ektlari ro'yxatini xossalaridan biri bo'yicha tartiblovchi C++ dasturini yozing.

9. C++ da sonlar juftligini yig'indisi bo'yicha saralashni amalga oshirish.

10. C++ tilida nomlar ro'yxatini uzunligi bo'yicha tartiblovchi funksiya yozing.

11. Tanlash tartiblash yordamida butun sonlar massivini tartiblovchi C++ dasturini yozing.

12. C++ tilida satrlar ro'yxati uchun qo'shishni tartiblash algoritmini amalga oshiring.

13. Haqiqiy sonlar massivini pufakchalar usuli yordamida tartiblovchi funksiyani C++ tilida yozing.

14. Sanalar ro'yxatini kamayish tartibida tartiblovchi C++ dasturini yozing.

15. C++ da mantiqiy qiymatlar massivida tanlash tartibini amalga oshirish.

16. C++ tilida talabalar ro'yxatini o'qishga kirgan yili bo'yicha tartiblovchi funksiyani yozing.

17. C++ tilida belgilar massivlarini kamayish tartibida tartiblashni amalga oshirish.

18. Bir nechta mezonlar asosida moslashtirilgan sinf ob'ektlari ro'yxatini tartiblovchi C++ dasturini yozing.

19. C++ tilida sonlar juftligini yig'indisining kamayish tartibida tartiblovchi funksiyani yozing.

20. C++ tilida satrlar ro'yxatini alifbo tartibida tartiblashni amalga oshirish.
21. Qo'shish usuli yordamida butun sonlar massivini tartiblovchi C++ dasturini yozing.
22. C++ tilida sanalar ro'yxati uchun tanlovni saralash algoritmini amalga oshiring.
23. Haqiqiy sonlar massivini tanlash usuli yordamida tartiblovchi funksiyani C++ tilida yozing.
24. Mantiqiy qiymatlar ro'yxatini o'sish tartibida tartiblovchi C++ dasturini yozing.
25. C++ tilidagi satrlar massivida qo'shish tartibini amalga oshirish.
26. C++ tilida o'quvchilar ro'yxatini yoshi bo'yicha tartiblovchi funksiyani yozing.
27. Belgilar massivini ASCII ortib borish tartibida tartiblovchi C++ dasturini yozing.
28. C++ da moslashtirilgan sinf ob'ektlari ro'yxatini ularning xossaligidan birining kamayish tartibida tartiblashni amalga oshirish.
29. C++ tilida sonlar juftligini ikkinchi elementning o'sish tartibida tartiblovchi funksiyani yozing.
30. Satrlar ro'yxatini uzunligi bo'yicha tartiblovchi C++ dasturini tuzing.