

20-Laboratoriya ishi

Элементлар жамланмасини бирор белги бўйича тартиблаштириш алгоритми

Маълумотlarni kompyuterda qayta ishlashda elementning informatsion maydoni va uning mashina xotirasida joylashishini bilish zarur. Shu maqsadda Маълумотlarni saralash amalga oshiriladi. Demak, saralash – bu маълумотlarni kalitlari bo'yicha doimiy ko'rinishda mashina xotirasida joylashtirishdan iborat. Bu yerda doimiylik маълумотlarni massivda kalitlari bo'yicha o'sishi tartibida berilishi tushuniladi.

Маълумотlarga qayta ishlov berilayotganda маълумотning informatsion maydonini hamda uning mashinada joylashishini (adresini) bilish zarur.

Saralashning ikkita turi mavjud: **ichki** va **tashqi**:

☐ -ichki saralash - operativ xotiradagi saralash;

- tashqi saralash – tashqi xotirada saralash.

Agar saralanayotgan yozuvlar xotirada katta hajmni egallasa, u holda ularni almashtirishlar katta sarf (vaqt va xotira ma'nosida) talab qiladi. Ushbu sarfni kamaytirish maqsadida, saralash kalitlar adresi jadvalida amalga oshiriladi. Bunda faqatgina маълумот ko'rsatkichlari almashtirilib, massiv o'z joyida qoladi. Bu usul adreslar jadvalini saralash usuli deyiladi. Saralanayotganda bir xil kalitlar uchrashi mumkin, bu holda saralangandan keyin bir xil kalitlilar boshlang'ich tartibda qanday joylashgan bo'lsa, shu tartibda qoldirilishi maqsadga muvofiq bo'ladi (Bir xil kalitlilar o'zlariga nisbatan). Bunday usulga turg'un saralash deyiladi.

Saralash samaradorligini bir necha mezonlar bo'yicha baholash mumkin:

☐ saralashga ketgan vaqt;

☐ saralash uchun talab qilingan operativ xotira;

☐ dasturni ishlab chiqishga ketgan vaqt.

Birinchi mezonni qarab chiqaylik. Saralash bajarilganda taqqoslashlar yoki almashtirishlar sonini hisoblash mumkin.

Faraz qilaylik, $N = 0,01n_2 + 10n$ – taqqoslashlar soni. Agar $n < 1000$ bo'lsa, u holda ikkinchi qo'hiluvchi katta, aks holda ya'ni, $n > 1000$ bo'lsa, birinchi qo'shiluvchi katta bo'ladi.

Demak, kichkina n larda taqqoslashlar soni n ga teng bo'ladi, katta n larda esa n_2 ga teng bo'ladi.

Saralashda taqqoslashlar soni quyidagi oraliqlarda bo'ladi:

dan gacha; – ideal holatda.

Saralashning quyidagicha usullari bor:

☐ qat'iy (to'g'ridan-to'g'ri) usullar;

☐ yaxshilangan usullar.

Qat'iy usullarning afzalliklarini ko'rib chiqaylik:

1. Bilamizki, dasturlarning o'zlarini ham xotirada joy egallaydi. To'g'ri, ridan-to'g'ri saralash usullarining dasturlari qisqa bo'lib, ular tushunishga oson.
2. To'g'ri, ridan-to'g'ri saralash usullari orqali saralash tamoyillarining asosiy xususiyatlarini tushuntirish qulay.
3. Murakkablashtirilgan usullarda uncha ko'p amallarni bajarish talab qilinmasada, ushbu amallarning o'zlarini ham ancha murakkabdir. Garchi yetarlicha katta n larda ulardan foydalanish tavsiya etilmasada, kichik n larda mazkur usullar tezroq ishlaydi.

Shu joyni o'zida qat'iy usullarni ishlash tamoyillariga ko'ra 3 ta toifaga bo'lish mumkin:

1. To'g'ri, ridan-to'g'ri qo'shish usuli (by insertion);
2. To'g'ri, ridan-to'g'ri tanlash usuli (by selection);
3. To'g'ri, ridan-to'g'ri almashtirish usuli (by exchange).

To'g'ri, ridan-to'g'ri qo'shish usuli bilan saralash algoritmi

Bunday usul karta o'ynida keng qo'llaniladi. Elementlar (kartalar) hayolan "tayyor" $a(1), \dots, a(i-1)$ va boshlang'ich ketma-ketliklarga bo'linadi. Har bir qadamda ($i=2$ dan boshlanib, har bir qadamda bir birlikka oshirib boriladi) boshlang'ich ketma-ketlikdan i -chi element ajratib olinib tayyor ketma-ketlikning kerakli joyiga qo'yiladi.

To'g'ri, ridan-to'g'ri qo'shish orqali saralash algoritmi quyidagicha bo'ladi:

```
for (int i=1; i<n; i++){
    x=a[i];
    x ni a[0]...a[i] oraliqning mos joyiga qo'shish
}
```

Kerakli joyni qidirish jarayonini quyidagi tartibda olib borish qulay bo'ladi. 2-elementdan boshlab har bir elementni qarab chiqamiz, ya'ni har bir element o'zidan oldin turgan element bilan solishtiriladi. Agar qaralayotgan element kichik bo'lsa, oldinda turgan element bilan o'rin almashadi va yana o'zidan oldinda turgan element bilan solishtiriladi, jarayon shu kabi davom etadi. Bu jarayon quyidagi shartlarning birortasi bajarilganda to'xtatiladi:

1. x elementi oldida uning kalitidan kichik kalitli $a(j)$ elementi chiqqanda.
2. x elementi oldida element qolmaganda.

```
for (int i=1; i<n; i++){ int j=i;
    while(a[j]<a[j-1]){
        int t=a[j-1];
        a[j-1]=a[j];
        a[j]=t;
        j=j-1;
    }
}
```

Algoritm samaradorligi

Faraz qilaylik, taqqoslashlar soni C , o'rinashtirishlar soni M bo'lsin. Agar massiv elementlari kamayish tartibida bo'lsa, u holda taqqoslashlar soni eng katta bo'lib, u ga teng bo'ladi, ya'ni n^2 . O'rinashtirishlar soni esa ga teng bo'ladi, ya'ni n . Agar

berilgan massiv oʻrnatilgan tartibida saralangan boʻlsa, u holda taqqoslashlar va oʻrnatilishlar soni eng kichik boʻladi, yaʼni .

Tanlash orqali saralash algoritmi

Mazkur usul quyidagi tamoyillarga asoslangan:

1. Eng kichik kalitga ega element tanlanadi.
2. Ushbu element birinchi element bilan oʻrnatiladi.
3. Keyin mazkur jarayon qolgan **n-1**, **n-2** elementlar bilan takrorlanib, to bitta eng “katta” element qolguncha davom ettiriladi.

```
for(int i=0;i<n-1;i++)
```

```
for(int j=i+1;j<n;j++)
```

```
if (a[i] > a[j]){
```

```
int k = a[j];
```

```
a[j]= a[i];
```

```
a[i]= k;
```

```
}
```

Algoritm samaradorligi:

- Taqqoslashlar soni

$$S = N(N-1)/2 = (N^2 - N)/2$$

- Massiv tartiblanganda oʻrnatilishlar soni

$$M_{\min} = 3(N-1)$$

- Massiv teskari tartiblanganda oʻrnatilishlar soni

$$M_{\min} = M_{\min}N/2 = 3N(N-1)/2$$

Ushbu usul boʻyicha saralash bajarilsa, eng yomon holda taqqoslashlar va oʻrnatilishlar soni tartibi n^2 boʻladi.

Pufaksimon saralash algoritmi

Ushbu usulning gʻoyasi quyidagicha: **n - 1** marta massivda quyidan yuqoriga qarab yurib kalitlar jufti-jufti bilan taqqoslanadi. Agar pastki kalit qiymati yuqoridagi jufti kalitidan kichik boʻlsa, u holda ularning oʻrni almashtiriladi (1- rasm).

Misol : massiv - 4, 3, 7, 2, 1, 6.

	1	2	3	4	5
1	4	3	3	3	3
2	3	4	4	4	4
3	7	7	2	2	2
4	2	2	7	1	1
5	1	1	1	7	6
6	6	6	6	6	7

1-rasm. Pufaksimon saralash usulida massiv elementlarining oʻrnini almashtirish

Pufaksimon usulni massiv elementlarida pastdan yuqoriga va yuqoridan pastga oʻtishni bir vaqtda amalga oshirish natijasida yaxshilash mumkin.

Taqqoslashlar soni:

$$M = (n/2)(n/2) = n^2/4$$

Almashtirishlar soni:

$$C_{\max} = 3n^2/4$$

1	2	3	4
4	2	1	--
5	4	2	--
2	1	3	--
1	3	4	--
3	5	5	--

2-rasm. Massivni pufaksimona saralashga misol

2-rasmda berilgan misolda 5 ta elementdan iborat massiv berilgan. Demak, massivda pastdan yuqoriga (yuqoridan pastga) o'tishlar soni $5-1=4$ marta bo'ladi. Misoldan ko'rinib turibdiki, algoritm ichki siklda 3-qadamdan boshlab massivni "bekor" qayta ishlaydi, 4-qadamni bajarmasa ham bo'ladi.

Berilgan usullarning afzalligi:

- 1) Eng sodda algoritm;
- 2) Amalga oshirish sodda;
- 3) Qo'shimcha o'zgaruvchilar shart emas.

Kamchiliklari:

- 1) Katta massivlarni uzoq qayta ishlaydi;
- 2) Har qanday holatda ham o'tishlar soni kamaymaydi.

"Pufaksimona" usulni yaxshilash

- 1) Agar massivda o'tishlar nafaqat yuqoridan pastga, balki bir vaqtning o'zida pastdan yuqoriga ham bo'lsa, u holda "yengil" elementlar "yuqoriga suzib" chiqadi va "og'ir" elementlar esa "cho'kadi".
- 2) Massivda "bekor" o'tishni yo'q qilish uchun, tashqi siklda massiv saralanganligini tekshiruvchi belgi qo'yish lozim.

```
for (int i=0; i<n; i++)
```

```
for (int j=n-1; j>i; j--)
```

```
if (a[j] < a[j - 1]){
```

```
int x = a[j - 1];
```

```
a[j - 1] = a[j];
```

```
a[j] = x; }
```

O'rinashtirish va taqqoslashlar soni: $(n * \log(n))$.

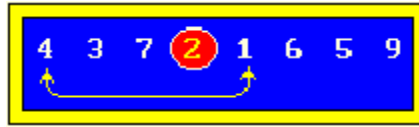
Quicksort – tez saralash algoritmi

Bu algoritm "bo'lib ol va egalik qil" tamoyilining yaqqol misolidir. Bu algoritim rekursiv bo'lib, o'rtacha $N * \log_2 N$ ta solishtirish natijasida saralaydi. Algoritm berilgan massivni saralash uchun uni 2 taga bo'lib oladi. Bo'lib olish uchun ixtiyoriy elementni tanlab undan 2 ta qismga ajratiladi. Lekin o'rtadagi elementni tanlab, massivning teng yarmidan 2 ga ajratgan ma'qul. Tanlangan kalit elementga nisbatan chapdagi va o'ngdagi har bir element solishtiriladi. Kalit elementdan kichiklar chapga, kattalar o'ng tomonga o'tkaziladi (6.3-rasm). Endi massivning har ikkala tomonida xuddi yuqoridagi amallar takrorlanadi. Ya'ni bu oraliqlarning o'rtasidagi elementlar kalit sifatida olinadi va h.k.

Misol uchun rasmdagi massivni saralash algoritmini ko'rib chiqamiz.

1. Oraliq sifatida 0 dan $n-1$ gacha bo'lgan massivning barcha elementlarini olamiz.

2. Oraliq o,,rtasidagi kalit elementni tanlaymiz, ya"ni
 $key = (\langle oraliq_boshi \rangle + \langle oraliq_oxiri \rangle) / 2$, $i = \langle oraliq_boshi \rangle$,
 $j = \langle oraliq_oxiri \rangle$.



3-rasm. Quicksort algoritmda o,,rinlashtirish

3. Chapdagi i -elementni **key** bilan solishtiramiz. Agar **key** kichik bo,,lsa, keyingi qadamga o,,tamiz. Aks holda $i++$ va shu qadamni takrorlaymiz.

4. O,,ngdagi j -element bilan **key** solishtiriladi. Agar **key** katta bo,,lsa, keyingi qadamga o,,tamiz, aks holda $j--$ va shu qadamni takrorlaymiz.

5. i - va j -elementlarning o,,rni almashtiriladi. Agar $i \leq j$ bo,,lsa, 3-qadamga o,,tiladi.

Birinchi o,,tishdan keyin tanlangan element o,,zining joyiga kelib joylashadi.

6. Endi shu ko,,rilayotgan oraliqda **key** kalitning chap tomonida elementlar mavjud bo,,lsa, ular ustida yuqoridagi amallarni bajarish lozim, ya"ni ko,,riladigan oraliq **0** dan **key-1** gacha deb belgilanadi va 2-qadamga o,,tiladi. Aks holda keyingi qadamga o,,tiladi.

7. Endi shu ko,,rilayotgan oraliqda **key** kalitning o,,ng tomonida elementlar mavjud bo,,lsa, ular ustida yuqoridagi amallarni bajarish lozim, ya"ni ko,,riladigan oraliq **key+1** dan **n-1** gacha deb belgilanadi va 2-qadamga o,,tiladi. Aks holda algoritm tugaydi.

Shu algoritmgaga misol ko,,rib chiqamiz.

Misol: Talabalar ism-sharifi va tartib raqamidan iborat jadvalni quicksort algoritmi bilan saralang va nechta o,,rinlashtirish amalga oshirilganini aniqlang.

Dastur kodi

```
#include <cstring>
#include <iostream>
using namespace std;
struct table{
    int t;
    string FIO;};
int q=0;
void qs(table *a,int first,int last){
    int i = first, j = last, table x = a[(first + last) / 2];
    do {
        while (a[i].FIO < x.FIO) i++;
        while (a[j].FIO > x.FIO) j--;
        if(i <= j) {
            if (i < j){ swap(a[i], a[j]);q++;}
            i++;
        }
    }
}
```

```

j--;
}
} while (i <= j);
if (i < last)
qs(a,i,last);
if (first < j)
qs(a,first,j);
}
int main(int args, char *argv[])
{ int n;cout<<"n=";cin>>n;
table talaba[n];
for(int i=0;i<n;i++){
talaba[i].t=i+1;
cin>>talaba[i].FIO;
}
qs(talaba,0,n-1);
for(int i=0;i<n;i++)
cout<<talaba[i].t<<" "<<talaba[i].FIO<<endl;
cout<<"quicksort algoritmi "<<q<<" ta o'rinlashtirishda saraladi\n";
system("PAUSE");
}

```

Dastur natijasi:

talabalar sonini kiriting=5
5 ta talabalar FIO sini kiriting
Farhod
Asror
Sobir
Bobur
Vali
/ 2 / Asror /
/ 4 / Bobur /
/ 1 / Farhod /
/ 3 / Sobir /
/ 5 / Vali /

Bu algoritm jadvalni 3 ta o'rinlashtirishda saraladi

Topshiriq

10. 12.2012 yilga qayta tekshirib olingan mashinalar, shuningdek, yangi mashinalar dasturini ishlab chiqish (variantga mos ravishda):

1. Mashina egalari ismlari bo'yicha alifbo tartibida joylashtirilsin va mos ravishda ularning mashinalari haqidagi ma'lumotlar chiqarilsin.

2. Avtomobillarni ta'mirlash tartibi ishlab chiqilsin. Bu yerda ta'mir tugashi sanasi qaysi avtomobil uchun ertaroq bo'lsa, shunga birinchi navbatda xizmat ko'rsatiladi.

3. Oldingi ta'mir qilinganlar soni 2 ga teng bo'lgan mashinalar raqamlari bo'yicha kamayish tartibida joylashtirilsin.

4. Oldin ta'mir qilinmagan mashinalarni ta'mirdan chiqish sanasi bo'yicha o'sish tartibida joylashtiring.

5. "Mercedes" markali mashina egalari alifbo bo'yicha teskari tartibda joylashtiring.

6. Boshqalaridan oldinroq ta'mirlanadigan mashinalarni ularning markasi bo'yicha alifbo tartibida joylashtiring (ta'mir tugatilishi sanasi 31.12.2012 dan erta).

7. "Nexia" markasidagi mashinalarni raqamlari bo'yicha o'sish tartibida joylashtiring.

8. O'tgan yildan beri ta'mirlanmagan mashinalarni ularning egalari ismlari bo'yicha alifbo tartibida joylashtiring.

9. Keyingi oyda ta'mirlanishi lozim bo'lgan mashinalarni oxirgi marta ta'mirlanganlik sanasi bo'yicha o'sish tartibida keltiring.

10. "Mercedes" markasidagi mashinalarni raqamlari bo'yicha kamayish tartibida joylashtiring.

N ta talabadan iborat guruh tuzilsin. Quyidagi ma'lumotlar berilgan: familiya, ism, tug'ilgan yili, fanlar bo'yicha bahosi: MTvaA, oliy matematika, fizika, dasturlash, topshirgan sessiya umumiy bali.

To'g'ri tanlov usulidan foydalanib, saralashni amalga oshirish dasturini ishlab chiqing (variantga mos ravishda):

11. Talabalar familiyalarini alifbo tartibida.
12. Talabalarni yoshi bo'yicha o'sish tartibida.
13. Talabalarni umumiy bali bo'yicha o'sish tartibida.
14. Talabalarni birinchi imtihoni natijasi bo'yicha o'sish tartibida.
15. Talabalarni ikkinchi imtihoni natijasi bo'yicha kamayish tartibida.
16. Talabalarni uchinchi imtihoni natijasi bo'yicha o'sish tartibida.
17. Talabalarni to'rtinchi imtihoni natijasi bo'yicha kamayish tartibida.
18. Talabalarni birinchi va ikkinchi imtihoni natijalari bo'yicha o'sish tartibida.

19. Talabalarni birinchi va ikkinchi imtihoni natijalari bo'yicha kamayish tartibida.

20. Talabalarni umumiy bali bo'yicha kamayish tartibida.

Pufaksimom saralash usulidan foydalanib, saralashni amalga oshirish dasturini ishlab chiqish (variantga mos ravishda):

21. A massivning eng katta (eng kichik) elementini ekranga chiqarish dasturini tuzing.

22. A massiv elementlari qiymatlarini kamayish tartibida saralash dasturini tuzing.

23. A massivda elementlar berilgan. Mazkur massiv elementlaridan shunday

V massiv shakllantiruvchi shunday dastur tuzingki, V massiv elementlari kamayish tartibida saralangan bo'lsin.

24. Elementlari o'sish tartibida joylashgan A sonli massiv va a soni berilgan. a ni A massivea shunday o'sishdagi tartiblanishlik buzilmasin

24. Elementlari o'sish tartibida joylashgan A sonli massiv va a soni berilgan a ni A massivga shunday qo'shinki, tartiblanganlik buzilmasin.

123

25. Elementlari o'sish tartibida joylashgan A massivni, elementlari kamayish tartibida joylashgan massiv ko'rinishida tez quruvchi dastur tuzing.

26. Manfiy va musbat sonlardan tashkil topgan A massiv berilgan. Barcha manfiy sonlarni chiqarib, musbatlarini o'sish tartibida joylashtiruvchi dastur tuzing.

27. Berilgan A massivdan ketma-ket sonlar olib, ulardan o'sish tartibida shakllantirilgan V massiv hosil qiluvchi dastur tuzing.

28. Mualliflar ro'yhati A massiv shaklida berilgan. Mualliflarni alifbo tartibida shakllantirish va shakllangan ro'yhatni ekranga chiqarish dasturini tuzing.

29. Telefon stansiyasida n ta mijoz bor. Quyidagi shaklda ro'yhat hosil qiluvchi dastur tuzing: telefon raqami, mijoz familiyasi (telefon raqamlari o'sish tartibida joylashadi).

30. A massivni uzunliklari har xil bo'lgan n ta so'z tashkil qiladi. So'zlarni uzunliklari bo'yicha o'sish tartibida joylashtiruvchi dastur tuzing.