7 - Mavzu: Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari. Saralash usullarini taqqoslash. Izlash algoritmlari.

1. O'rniga qo'yish bilan saralash algoritmi

Ushbu saralash algoritmining asosiy mohiyati saralangan ro'yxatga yangi element qo'shishda uni “o'z joyiga” joylashtirishdan iboratdir. Bunda algoritm saralanuvchi ro'yxat birinchi elementini uzunligi 1 ga teng bo'lgan saralangan ro'yxat deb qabul qilib, ikkinchi elementni yangi yaratilayotgan saralangan ro'yxatning “kerakli” joyiga joylashtiradi. So'ngra berilgan ro'yxatning uchinchi elementi ham saralangan ikki elementli ro'yxatdagi o'z joyiga joylashtiriladi va hokazo. Ushbu jarayon berilgan ro'yxatning barcha elementlari saralangan ro'yxatga joylashtirib chiqilgunga qadar davom ettiriladi. O'rniga qo'yish algoritmining ifodasi quyidagidan iborat:

```
InsertSort(List,N)
For i=2 to N do
newElement=list[i]
location=i-1
while (location) >=1) and(list[location]> newElement) do
list[location+1] = list[location]
location= location-1
end while
list[location+1] = newElement
end For
```

Ushbu algoritm newElement o'zgaruvchisiga yangi o'rniga qo'yiluvchi qiymatni kiritadi. So'ngra bu yangi elementga joy ajratish uchun massiv elementlari bir pozitsiyaga suriladi (while sikli). Siklning oxirgi iteratsiyasi location+1 nomerli elementni location+2 pozitsiyaga o'tkazadi,ya'ni location+1 pozitsiyasi yangi element uchun bo'shatiladi.

2.Eng yomon holat tahlili

While siklida amallar eng ko'p bajariladi, qachonki ro'yxatning saralangan qismiga yangi qo'shiluvchi element bu ro'yxat elementlarining barchasidan kichik bo'lsa. Bu holatda sikl location o'zgaruvchisining qiymati 0 ga teng bo'lganda to'xtaydi.Shuning uchun har bir yangi element ro'yxatning boshidan joy olsa, algoritm eng ko'p ish bajaradi. Bu faqat berilgan ro'yxat elementlari kamayib borish tartibida joylashgan bo'lgandagina mumkindir.Bu holat eng yomon holatlardan biridir.Bunday ro'yxatni qayta ishlash jarayoni quyidagicha aalga oshiriladi: birinchi bo'lib berilgan ro'yxatning ikkinchi elementi joylashtiriladi.U faqat bitta element bilan taqqoslanadi. Ikkinchi joylashtiriluvchi element oldingi ikkitasi bilan taqqoslanadi, uchinchisi esa oldingi uchtasi bilan va hokazo i - element oldingi i ta element bilan taqqoslanadi. Shunday qilib, jarayon N - 1 marta takrorlanadi. O'rniga qo'yishlar bilan saralash algoritmining murakkabligi eng yomon holat uchun quyidagicha hisoblanadi.

3.O’rtacha holat tahlili

Ushbu tahlil jarayonini ikki etapga ajratamiz. Oldiniga navbatdagi elementning pozitsiyasini aniqlash uchun zarur bo’lgan taqqoslashlar o’rtacha sonini hisoblaymiz. So’ngra ushbu miqdordan foydalanib barcha amallarning o’rtacha qiymatini hisoblash mumkin.Bitta element uchun mumkin bo’lgan pozitsiyalar nechtagacha bo’ladi? Saralangan ro’yxatga qo’shiluvchi birinchi element uchun ikkita umkin bo’lgan holat mavjud: u ikki elementli ro’yxatda birinchi yoki ikkinchi pozitsiyani egallaydi. Ikkinchi elementda uchta mumkin bo’lgan holat bo’ladi:birinchi, ikkinchi yoki uchinchi. Demak, i-element mumkin bo’lgan i + 1 ta pozitsiyadan birini egallaydi. Faraz qilaylik, bu imkoniyatlarning ehtimoli o’zaro teng bo’lsin. U holda i-

elementni saralangan ro’yxatga qo’shish uchun bajariladigan barcha amallarning o’rtacha qiymati quyigi formular bilan hisoblanadi:

4.Pufakchali saralash

Pufakchali saralash algoritmining mohiyati kichik qiymatlarning ro’yxat yuqorisiga itarilib, yirik qiymatlarning ro’yxat pastiga surilishiga asoslangan. Pufakchali saralashning ko’p variantlari mavjud bo’lib, ulardan birini ko’rib o’tamiz. Bunda algoritm ro’yxat bo’ylab bir necha o’tishni bajaradi. Har bir o’tishda qo’shni elementlar bir-biri bilan taqqoslanadi.Agar bu elementlarni tartibi noto’g’ri bo’lsa, ularning o’rinlari almashtiriladi.Har bir o’tish ro’yxat boshidan boshlanadi. Oldin birinchi va ikkinchi element taqqoslanadi, keyin ikkinchi va uchinchisi va hokazo. Bunda ro’yxatning eng katta elementi birinchi o’tish tugagandan keyin ro’yxatning oxiridan joy oladi.Ikkinchi o’tishda kattalik bo’yicha ikkinchi element ixiridan ikkinchi o’rinni egallaydi. Agar biror o’tishda bitta ham o’rin almashtirish bajarilmasa, bro’yxat saralangan deb hisoblanib, algorit ishi to’xtatiladi. Quyida pufakchali saralash algoritmining ifodasi keltirilgan:

5.O’rtacha holat tahlili

Eng yomon holatda For sikli N – 1marta takrorlanishini ko’rdik. O’rtacha holatda almashtirishsiz o’tishlarning bajarilish ehtimoli barcha N - 1 ta o’tish uchun teng deb hisoblayiz.Har bir o’tishda nechta taqqoslash bajarilishini ko’raylik.Birinchi o’tishdan keyingi to’xtashdan keyin taqqoslashlar soni N – 1 ga teng.Ikkita o’tishdan keyin taqqoslashlar soni N - 1 + N – 2 ga teng. S(i) bilan birinchi i ta o’tishdan jarayonida bajarilgan taqqoslashlar sonini belgilaymiz.

6.Piramidali saralash algoritmi

Piramidali saralash algoritmining asosida binar daraxtning piramida deb ataluvchi maxsus turidan foydalanish yotadi. Bunday binar daraxt tugunlarining qiymati eng yaqin avlodlari qiymatidan doimo katta bo’ladi.Saralash jarayoni piramida qurilishidan boshlanadi. Bunda ro’yxatning aksimal elementi daraxtning eng yuqori tugunida joylashadi. So’ngra ushbu element ro’yxatning eng oxirgi navbatiga joylashtiriladi.Elementi olingan piramida esa qaytadan quriladi.Natijada daraxt ildizida kattalik bo’yicha ikkinchi o’rinda turadigan element joylashadi va uni ro’yxatning oxiridan bitta oldingi o’ringa o’tkaziladi.Protsedura barcha elementlar ro’yxatdagi o’z o’rinlarini egallagunlaricha davom etadi.

7.Piramidani qayta qurish

Piramidaning ildizi ro’yxatga ko’chirilganda, ildiz element bo’sh qoladi. Uning joyiga avlod elementlaridan kattasi joylashtirilishi kerak. Piramidani qayta shakllantirish jarayoni eng quyi darajaning o’ngdan birinchi elementidan boshlanadi. Natijada piramida quyi darajasidagi elementlar bir tekis yo’qotib boriladi:
Bu erda root o’zgaruvchisining vazifasi nimada?,- degan savol tug’iladi. Ushbu qo’shimcha parametr piramidani pastdan yuqoriga qurish imkonini beradi.

8.Tanlash algoritmi

Ba'zi holatlarda bizga ro’yxatdagi konkret qiymatga ega bo’lgan elementni emas, balki boshqa xususiyatga ega bo’lgan elementni izlashga to’g’ri keladi. Masalan, yozuv sohalarining kattalik bo’yicha k-o’rinda turgan qiymatini topish talab etilsin. Bunday yozuvni topishning usullaridan biri ro’yxatni kamayish tartibidan saralashdan iborat; bunda kattaliu bo’yicha k-yozuv k-o’ringa joylashtiriladi. Bu usul keragidan ko’proq mehnat talab qiladi: chunki izlangandan kichik bo’lgan elementlar bizni qiziqtirmaydi.Bu vaziyatdan chiqishning yana bir usuli mavjud: ro’yxatdan eng katta element topilib, ro’yxatning oxiriga joylashtiriladi.So’ngra ro’yxat qolgan

qismining eng katta elementi topiladi va ro’yxat oxiridan ikkinchi o’ringa joylashtiriladi. Ushbu protsedurani K marta takrorlab, kattalik bo’yicha K-o’rinda turuvchi elementni topib olamiz:

9.Eng yomon holat tahlili

Algoritm Piramida protsedurasi asosida qurilganligi uchun, ishni uning tahlilidan boshlaymiz. Piramidaning har bir qatlamida algoritm ikki eng yaqin avlodni taqqoslab, ulardan kattasini kalit bilan taqqoslaydi.Bundan chu qurligi Dga teng b o’ lgan piramida uchun ta q qoslashlar soni 2D2 dan oshmasligi kelib chiqadi.Piramidani shakllantirish qadamida Piramida protsedurasi ikkinchi qatlamning oxiridan boshlab har bir tugun uchun chaqiriladi, ya'ni buna piramidalarning chuqurligi 1 ga teng bo’ladi.So’ngra ushbu protsedura uchinchi qatlamning oxiridan boshlab har bir tugun uchun chaqiriladi va chuqurligi 2 ga teng bo’lgan piramidalar quriladi.Oxirgi o’tishda ildiz darajasidagi shakllantirilgan piramidaning chuqurligi ga teng bo’ladi. Endi Piramida protsedurasining har bir o’tishdagi tugunlar sonini hisoblash kerak. Ildiz qatlamida tugun bitta, ikkinchi qatlamda uning ikki avlodi joylashadi, uchinchi qatlamda t o’ rtta va hokazo

10.O’rtacha holat tahlili

Ishni boshlang’ich massiv teskari tartibda joylashgan eng yaxshi holatdan boshlaymiz. Elementlarning bunday joylashuvi bizga avtomatik holda to’g’ri piramidani beradi. Shuning uchun har bir Piramida protsedurasiga murojaat vaqtida elementlarning to’g’ri joylashganligini tasdiqlovchi ikkita taqqoslash amali bajariladi. Ushbu protsedura elementlarning taxminan yarmi uchun chaqirilganligidan piramida qurish mobaynida N ga yaqin taqqoslash amallari bajarilishi kelib chiqadi. Saralangan massivga ega bo’lish uchun piramidadan barcha elementlarni ketma-ket olib, uni har safar qaytadan shakllantirish lozim.Shuning uchun eng yaxshi holatda piramidali saralash algoritmi N ga teng b o’ladi.Shunday qlib, piramidali saralash algoritmining eng yaxshi holat murakkabligi bilan eng yomon holat murakkabligi mos tushadi.

11.Birlashtirish bilan saralash algoritmi

Ushbu saralash algoriti rekursiv xarakterga egadir. Bitta elementdan iborat bo’lgan ro’yxat saralangan bo’lganligi uchun algoritm ro’yxatni bir elementli ro’yxatlarga ajratib, so’ngra ularni ketma-ket birlashtiradi.Bunda ro’yxat ketma-ket ikki qismga ajratib boriladi. Ikkiga bo’lish jarayoni bo’lak ro’yxat birinchi elementining nomeri shu bo’lakdagi oxirgi elementi nomeridan kichik bo’lgunga qadar davom etadi. Agar navbatdagi bo’lakda bu shart bajarilmasa, bitta elementli bo’laklar hosil bo’lgan deb hisoblanadi. Uzunligi birga teng bo’lgan ro’yxatlar uchun ikkita murojaatdan so’ng ushbu iki ro’yxatni birlashtiruvchi protsedura chaqiriladi. Buning natijasida uzunligi ikkiga teng bo’lgan saralangan ro’yxat hosil bo’ladi. Keyingi bosqichda ikkiga teng uzunlikdagi saralangan ro’yxatlar uzunligi to’rtga teng bo’lgan saralangan ro’yxatlarga birlashadi. Bu jarayon ikkita saralangan yarim ro’yxatlar birlashuvchi qadamgacha davom etadi.

12.MergeLists algoritmining tahlili

MergeLists protsedurasi elementlarni taqqoslash vazifasini bajaradi. A ro’yxatning barcha elementlari V ro’yxatning barcha elementlaridan kichik bo’lgan holni ko’ramiz. Bunda A[1] va B[1] elementlar taqqoslanadi va A[1] element kichik bo’lganligi uchun S ga joylashtiriladi. So’ngra B[1] va A[2] elementlar taqqoslanib, A[2] element S ga joylashtiriladi.A ro’yxat elementlarining B[1] bilan taqqoslanishlar soni NA A ro’yxat elementlari soniga teng bo’ladi. Aks holda, agar V ro’yxatning barcha elementlari A ro’yxat elementlaridan kichik bo’lsa, taqqoslashlar soni NV V ro’yxat elementlari soniga teng bo’ladi. A ro’yxatning 1-elementi B ro’yxat 1-elementidan katta bo’lib, A ro’yxatning barcha elementlari V ro’yxatning 2-elementidan kichik bo’lganda A[1] va V[1] taqqoslanib, V[1] S ro’yxatga o’tkaziladi.So’ngra A ro’yxatning

qolgan elementlari V[2] bilan taqqoslanib, ketma-ket S ga o'tkaziladi. Bunda taqqoslashlarning to'liq soni N A + 1 ga teng bo'ladi. Bundan birinchi situatsiyaning eng yaxshi holat ekanligi kelib chiqadi.

Quyidagi situatsiya ushbu algoritm uchun eng yomon holat bo'lib hisoblanadi: A[1] element V[1] va V[2] orasida, A[2] element B[2] va B[3] orasida, A[3] element B[3] va B[4] orasida bo'ladi va hokazo. Bunda S ro'yxatga ko'chirib o'tkazish har ikkala ro'yxatdan navbat bilan amalga oshiriladi. Bunda har ikkala ro'yxat elementlari uchun NA va NB ga teng taqqoslashlar amalga oshirilganligi uchun eng yomon holat uchun algoritm murakkabligi NA+ NB-1 ga teng bo'ladi.

13.MergeSort algoritmining tahlili

Ushbu funktsiya first o'zgaruvchisining qiymati last o'zgaruvchisining qiymatidan kichik bo'lgan holda chaqiriladi. Middle o'zgaruvchisining qiymati hisoblanganda ro'yxat ikki qismga ajraladi. Middle o'zgaruvchisining qiymati first va last o'zgaruvchilari qiymatlarining o'rtasida joylashganligi uchun ro'yxat ikki teng qsmga ajraladi.Har bir qism ro'yxat N/2 ta elementdan iborat bo'ladi. Bunda MergeLsits algoritmning tahliliga ko'ra, birlashishi jarayonida eng yaxshi holatda N/2 ta taqqoslash amali bajariladi. Bundan birlashtirish bilan saralash algoritmining murakkabligi eng yaxshi va eng yomon holatlar uchun bir xilda ekanligini keltirib chiqarish mumkin.

14.Tez saralash algoritmi

Tez saralash yana bitta rekursiv algoritmdir.Uning ma'nosi quyidagicha: ro'yxatdan biror elementni tanlab, algoritm uning yordamida ro'yxatni ikki qismga ajratadi. Birinchi qism ro'yxatga ushbu elementdan kichik qiymatlar, ikkinchisiga ushbu elementdan kattalari joylashtiriladi. Keyingi qadamda ushbu ikki qism ro'yxat xuddi shu usul bilan yana ikki qismga ajratiladi va hokazo. Bunda jarayon har bir qism ro'yxat bitta elementdan iborat bo'lgunga qadar davom ettiriladi

15.Ro'yxatni bo'laklarga ajratish

PivotList funktsiyasi namuna element sifatida ro'yxatning birinchi elementini olib, pivot ko'rsatkichini ro'yxat boshiga o'rnatadi.So'ngra u ro'yxatning qolgan elementlarini namuna bilan taqqoslaydi. Namunadan kichik element topilganda PivotPoint ko'rsatkichini oshirib, topilgan elementni PivotPointning yangi nomeridagi element joyini almashtiradi. Ro'yxatning biror qismi elementlari tekshirib bo'linganda, ro'yxat to'rt qismga ajralib qoladi: birinchi qism birinchi namuna elementdan; ikkinchi qism first +1 pozitsiyadan boshlanib, PivotPoint ko'rsatkichi pozitsiyasida tugaydigan barcha tekshirilgan elementlardan tashkil topadi; uchinchi qism PivotPoint +1 pozitsiyadan boshlanib, index sikli parametrining ko'rsatkichi qiymati bilan tugaydi; ro'yxatning qolgan qismi hali tekshirilmagan eleentlardan tashkil topadi

16.Eng yomon holat tahlili

PivotList protsedurasi N elementdan iborat ro'yxat uchun chaqirilganda , u N – 1 ta tqqoslash amalini bajaradi, chunki PivotValue ning qiymati ro'yxatning qolgan barcha eleentlari bilan taqqoslanadi.Eng yaxshi holatda PivotList ro'yxatni teng uzunlikdagi ikki bo'lakka ajratadi.Eng yomon holatda esa ushbu bo'laklar uzunligi bir-biridan maksimal darajada farq qilishi kerak. Bunga PivotValue qiymati ro'yxatning qolgan barcha elementlaridan katta yoki kichik bo'lganda erishish mumkin.Bunda ro'yxatlarning birida bitta ham element bo'lmaydi, ikkinchisi esa N - 1 elementdan tashkil topadi. Agar har bir rekursiv murojaatda bunday holat yuz beradigan bo'lsa, har safar bitta elementga kamayish yuz beradi

17.O’rtacha holat tahlili

Algoritm ishida asosiy vazifani PivotList prtsedurasi bajarganligi uchun uning ishini tahlil qilamiz. PivotList algoritmi bajarilgandan keyin N ta pozitsiyadan ixtiyoriysi PivotValue ni o’zida saqlashi mumkin. Eng yomon holat tahlilida PivotList protsedurasi N elementdan iborat ro’yxatni bo’laklarga bo’lib, N – 1 ta taqqoslash amalini bajarishini ko’rdik.Ushbu ro’yxatlarni birlashtirish hech qanday xarakatni talab etmaydi. PivotList protsedurasi R qiymatni berganda R – 1 va N – R uzunlikdagi ro’yxatlar uchun Quicksort protsedurasiga murojaat yuz beradi. O’rtacha holat tahlilida R ning barcha mumkin bo’lgan
N ta qiymatlari hisobga olinishi kerak

18.Diskli xotira qurilmasining tuzilishi

Tashqi saralash jarayoni tashqi xotirada saqlanuvchi axborotlarni saralash vazifasini bajaradi. Tashqi saralash jarayoni ichki saralashdan katta farq qiladi. Chunki tashqi fayllarga murojaat to’g’ridan – to’g’ri emas, ketma-ket(blolab) usulda amalga oshiriladi. Bunda axborotlarni faqat bloklab o’qish mumkin. Tashqi saralash jarayonini tushunish uchun disklarning fizik tuzilishi bilan umumiy tanishib chiqish kerak bo’ladi. Disklarning tashqi sirti magnitli qoplamga ega bo’lib, ular doimiy katta tezlik bilan o’z o’qi atrofida aylanadi. Disklarning har bir ish sohasiga bitta o’qish-yozish qurilmasi o’rnatilgan. Axborotlarga murojaat vaqtida o’qish-yozish qurilmasi tomonidan treklar deb ataluvchi diskdagi ma’lumotlar yozilgan yo’lakchalardan berilganlar o’qiladi. Bu treklar yig’indisi joriy silindr deb ataladi. qish-yozish qurilmalari maxsus shtangaga o’rnatilgan bo’lib, bu shtanga burilganda o’qishqyozish qurilmasi boshqa silindrga o’tqaziladi. Silindrlar shtanganing bir tomonga harakati vaqtida o’qish-yozish qurilmalari blokiningularga murojaat qilish tartibida nomerlanadi. Berilganlar elementlari orasidagi masofa ulr joylashgan silindrlar nomerlari farqidan iborat bo’ladi.Xotira elementlarini adreslash ular joylashgan silindr doirasida amalga oshiriladi. Fayl adreslar tartibi bo’yicha yoziladi, ammo bo’sh joy bo’lmaganda, boshqa silndrga ham yozilishi mumkin.Diskadagi axborotlarga murojaat asosiy xotiradagi axborotlarga murojaatdan anchagina sekin amalga oshiriladi.Chunki bunday murojaat vaqti bu jarayonda bir necha bajariladigan amalarga ketadigan vaqtdan kelib chiqadi:

1. Silindr kerakli elementining o’qishqyozish qurilmasi tagidan o’tishini ko’ish vaqti;
2. O’qish-yozish qurilmasining boshqa silindrga o’tqazilishini ko’ish vaqti;
3. Tashqi saralash vaqti;
4. Tashqi saralash vaqti ham o’z navbatida bir nechta amallar bajarilishiga ketadigan vaqtdan hosil bo’ladi:
5. Fayl qismlarining ichki saralanishi;
6. Berilganlarning ko’r marta diskka yozilishi va o’qilishi;
7. O’qish-yozish aktlari orasidagi golovka yurishlari;
8. Tartiblangan qismlarning birlashuvi jarayonidagi xotiradagi harakatlar;

Tashqi xotiradagi fayllarni saralash muhim amaliy ahamiyatga egadir. Bunday saralash jarayoni natijasida tashqi xotiradagi axborotlarga murojaat vaqti sezilarli kamaytiriladi va xotiraga axborotlar o’qish-yozish jarayoni ancha tezlashadi. Tashqi xotiradagi fayllarni saralash fayl bloklari utida bajariladi. Bunday saralash algoritmlaridan biri **Birlashuv** yo’li bilan saralash algoritmidir. Birlashuv tushunchasi ikki yoki undan ortiq tartiblangan ketma-ketliklarning bitta tartiblangan ketma-ketlikka ayni paytda joriy elementlarni siklik tanlash yordamida almashtirish(keltirish) ni bildiradi.Birlashuv jaryoni saralash jarayonlari ichida eng sodda jarayon hisoblanadi. Tashqi saralash usullarining katta qismi quyidagi umumiy tamoyillarga rioya qiladi. Saralanuvchi fayl bo’ylab birinchi u o’tishda xajmi taxminan operativ xotira xajmiga mos keluvchi bloklarga ajratiladi. Keyinchalik ushbu fayl bloklari saralanadi. Songra saralangan bloklarning birlashuvi amalga oshiriladi. Bu maqsadda bir necha o’tishlar amalga oshirilib, har bir o’tish jarayonida saralanganlik darajasi ortib boradi va jarayon fayl to’la saralanib bo’lgunga qadar davom ettiriladi. Bunday yondashuv **birlashuvli saralash** db ataladi. Birlashuvli saralash jarayonini realizasiya qiluvchi bir nechta algoritm mavjud. Bulardan biri **Bouz-Nelson** algoritmidir.

19.Ketma-ket qo’shib olish usulida saralash

Algoritm bir nechta fayl qismlariga ega bo’lgan holda, shulardan ikkitasini birlashtirishdan boshlanadi. So’ngra qolgan qismlar ham ketma-ket tartiblangan qismga qo’shib olinadi. Ushbu jrayon quyidagi etaplardan iborat:

Qo’shib olinishdan oldin navbatdagi fayl qismi xotiraning maxsus «A» sohasiga chaqiriladiva shu erda saralanib,qoldiriladi. Faylning oldin tartiblangan qismining boshi «B» sohaga chaqiriladi va birlashtirish jarayoni bajariladi.Bunda «B» sohadagi axborotlar davriy ravishda to’ldirib turiladi.bunda birlashuv natijalari «S» sohaga ketmaqket uzatiladi. «S» sohadan tartiblangan fayl qismi faylning ayni paytda qo’shib olinuvchi qismi joyiga joylashtiriladi.Ushbu jarayonda etaplar soni kam bo’lib, faylning katta bo’lmagan xajmlarida tez bajariladi

.

20.Takrorlanuvchi balansli birlashuv usuli

Bu saralash usulining birinchi etapida ichki saralash usullaridan foydalanib, faylning M ta tartiblangan katta R xajmli qimslari yaratiladi.Ularga nisbatan To’g’ri birlashuv algoritmi qo’llaniladi.Bunda qo’shimcha disk sohasi ajratilib, bu soha bevosita birlashuvchi qfayl qismlaridan oldin yoki keyin joylashtiriladi.Birlashuv jarayoni tugallangach,bu soha navbatdagi fayl qismlariga o’tqaziladi. Bu soha xajmi fayl qismlari xajmidan kam bo’lmaydi. Bunda ikki fayl qismining birlashuvi natijasi birinchi fayl qismi bilan rezerv xotira qismiga joylashtiriladi.Ikkinchi fayl qismining joyi esa bo’shaydi.Ushbu bo’shagan joy keyingi birlashuvchi fayl qismlari uchun rezerv vazifasini bajaradi.Birlashuv jarayoni natijasida rezerv xotiraqismi fayl boshidan fayl oxiriga siljib boradi va aksincha. Saralash jarayoni davomida rezerv xotira qismi kattalashib boradi, chunki birlashuvchi qismlar ning xajmi ortib boradi.

21.Ketma-ket izlash algoritmi va uning tahlili

Ro’yxatidan kerakli axborotni izlash nazariy programmalashning fundamental masalalaridan biridir. Izlash algoritmlari to’g’risida gap ketganda axborot dasturdagi berilganlar massividan iborat bo’lgan yozuvlarda saqlanadi degan nuqtai nazardan kelib chiqiladi.Yozuvlar yoki ro’yhat elementlari massivda ketma-ket joylashadi. Ko’pincha yozuvlar bir necha sohalardan iborat bo’lishi mumkmn, ammo izlashda ushbu sohalardan faqat bittasi(kalit) hisobga olinadi.Yozuvlar kalit sohasi qiymati bo’yicha saralangan yoki saralanmagan bo’lishi mumkin.

Izlash algoritmlari quyidagi guruxlarga bo’linadi:

a. Kalitlarni qiyoslashga asoslangan
b. Kalitlarning raqamli xususiyatlariga asoslangan

Saralangan ro’yxatda yozuvlar kalitning o’sib borish tartibida joylashgan bo’lib, saralanmagan ro’yxatda ular tasodifiy ravishda joylashadi.Saralanmagan ro’yxatdagi izlash jarayoni kerakli axborotga duch kelinmaguncha barcha yozuvlarni ko’rib chiqishdan iborat bo’ladi. Bu eng sodda izlash algoritidir. Konkret qiymatni izlash tanlash masalasi bilan bog’liq bo’lib, bunda qandaydir xususiyatga ega bo’lgan elementni topish talab etiladi. Masalan, kattalik bo’yicha beshinchi o’rindagi element, ro’yxat oxiridan ettinchi element yoki o’rtacha qiymatli element.

Izlash algoritmlarida ro’yxatni maqsad elementi deb ataluvchi qandaydir konkret elementni topishga qaratilgan ko’rib chtqish jarayoni amalga oshiriladi. Ketma-ket izlashda ro’yxat elementlari saralanmagan deb qabul qilinadi. Izlash jarayonida kerakli elementning ro’yxatda mavjud ekanligi tekshirilibgina qolmay, balki ushbu kalitga bog’liq bo’lgan ma'lumotlarga ham murojaat qilinadi.Masalan, kalitning qiymati xizatchining tartib nomeridan yoki boshqa identifikatordan iborat bo’lishi mukin. Kerakli kalit topilgandan so’ng dastur u bilan bog’liq ma'lumotlarni o’zgartirishi yoki bosmaga chiqarishi mumkin. Umuman olganda, izlash algoritmining maqsadi kalitning pozitsiyasini (turgan joyini) aniqlashdan iborat. Agar kalit qiymat topilmasa, algoritm massivning yuqori chegarasidan katta bo’lgan indeks qiymatini chiqaradi. Ketma-ket izlash algoritmi ro’yhat elementlarini birinchi elementdan boshlab, keraklisi topilmagunga qadar birma-bir ko’rib chiqadi. Kalitning konkret qiymati ro’yxatda qanchalik uzoq joylashgan bo’lsa, izlashga shunchalik ko’p vaqt sarflanadi.

22.Ikkilik izlash algoritmi va uning tahlili

Saralangan massivda biror elementni izlash jarayonida maqsad elementni massiv o’rtasidan olingan element bilan taqqoslaganda 3 ta holatdan biri yuz beradi: qiymatlar teng; maqsad element kichik; maqsad element katta. Birinchi holat eng yaxshi hisoblanib, izlash jarayoni to’xtaydi. Qolgan ikkila holatda ham massivning yarmini tashlab yuborish mumkin.Maqsad qiymat o’rtanchi elementdan kichik bo’lsa, u ro’yxatda o’rtancha elementdan oldin keladi, aks holda ushbu elementdan keyin keladi.Shu jarayonni davom ettirib, qro’yxatning qolgan qisining ha yarini tashlab yuboraiz va hokazo.

Foydalanilgan adabiyotlar

1. Informatika va informatsion texnologiyalar, M. Aripov va boshqalar. Oliy o’quv yurti talabalari uchun darslik. Toshkent-2019 y.
2. Axborot texnologiyalari, M. Aripov va boshqalar. Oliy o’quv yurti talabalari uchun o’quv qo’llanma. Toshkent-2019 y.
3. Delphi tilida dasturlash asoslari, Sh. Nazirov. Toshkent-2018 y.