

6-MA'RUZA. TANLASH VA JOYLASHTIRISH TURKUMIDAGI MURAKKABLIKKA EGA SARALASH ALGORITMLARI

Hisoblash texnikasi sohasidagi ko'plab olimlar saralashni algortimlarni o'rganishda eng fundamental masala deb qarashadi. Buning bir qancha sabablari bor:

1) Ba'zan ilovalarda axborotlarni saralashlikning aslo iloji bo'lmaydi. Masalan, bankda mijozlarning hisob holatlari to'g'risidagi hisobotni tayyorlash uchun ularning chek nomerlari bo'yicha saralashni bajarish kerak bo'ladi.

2) Odatda saralash algoritmlarda kalit qismdasturi sifatida ishlatiladi. Masalan, dasturda, turli xil darajalarda bo'lgan grafik obyektlarni vizual berkitishni bajarishda, dastlab ushbu obyektlarni chiqish tartibini o'rnatish uchun obyektlarni "pastdan tepaga" darajalari bo'yicha saralash kerak bo'ladi.

3) Saralash algoritmlarning ko'plab turli xil texnologiyalar qo'llaniladigan turlari mavjud. Algoritmlarni saralashda turli xil algoritm sinflariga qo'llaniladigan juda ko'p muhim usullardan foydalaniladi.

4) Algoritmlarni saralash jarayonini amalga oshirishda oldingi qatorga juda ko'plab amaliy muammolar chiqadi. Ko'plab saralash dasturi ishlab chiquvchilarni tanlash shu yoki boshqa vaziyatlarda juda ko'p faktlarga bo'g'liq bo'lishi mumkin. Juda ko'p shunga o'xshash masalalar hal qilishda "kodlarni sozlash" dan ko'ra "algoritmlar" darajasidan foydalanish maqsadga muvofiqdir.

Saralash algoritmlari:

– O'sish yoki kamayish tartibida to'plam elementlarini tartiblangan saralash deyiladi.

– Tartiblangan elementlar bilan ishlash tartibsiz joylashgan elementlardan ko'ra qulayroq: kerakli elementlarni yengil topish, olib tashlash, yangilarini qo'yish mumkin.

– Saralash algoritmlarini 6.1-rasmdagidek guruhlariga ajratish mumkin.

– Odatda saralanayotgan to'plam elementlari yozuvlar deyiladi va $k_1, k_2, \dots, k_n$ ko'rinishida yoziladi.

Umuman olganda saralashning maqsadi berilgan ob'yektlar to'plamini aniq bir tartibda guruhlab chiqish jarayoni tushuniladi.

Saralashning maqsadi keyinchalik, saralangan to'plamni qidirilayotgan elementini topishdan iborat. Bu qariyb universal, fundamental jarayon. Biz bu jarayon bilan har kuni uchrashamiz – telefon daftaridagi saralash, kitoblar sarlavhasida, kutubxonalarda, lug'atlarda, pochtada va h.k. Hatto yosh bolalar ham o'z narsalarini tartiblashga o'rganadi.

Saralashning juda ko'p usullari mavjud. Ular turli to'plamlar uchun turlicha bo'lishi mumkin.

Massivni saralash uchun ishlatiladigan usul unga berilgan xotirani ixcham holda ishlatish lozim. Boshqacha qilib aytganda, saralanayotgan massiv xuddi shu massivni o'zida amalga oshirilishi lozim.

Saralash algoritmlari kam mashina vaqtini talab qilishi lozim. Eng yaxshi tez algoritmlar $n * \log n$ tartibidagi saralashlarni talab etadi.

Bu yerda ikkita saralashni farqlash lozim. Ichki va tashqi saralash. Ichki saralash deganda, massivlarni saralashni, tashqi saralash deganda esa, fayl elementlarini saralashni tushunamiz.

Faraz qilaylik, bizga $a_1, a_2, \dots, a_n$ lementlar berilgan bo'lsin, u holda massivni saralash deganda, uni elementlarini o'rinlariga almashtirish tushuniladi. $a_{k_1}, a_{k_2}, \dots, a_{k_n}$, bu yerda, quyidagi tartiblashtirilgan funkstiya bajariladi:

$$f(a_{k_1}) \quad f(a_{k_2}) \quad \dots \quad f(a_{k_n})$$

Saralash algoritmlari bajarilish tezligida xotirani effektiv ishlatilishi bo'yicha baholash.

Vaqt – bu algoritmni tezligini xarakterlovchi asosiy parametr. Bu albatta hisoblash murakkabligi bilan bog'liq. Vaqt bo'yicha algoritmlarni uchta turga bo'lish mumkin: yomon, o'rta va yaxshi.

- **$O(n*n)$** – yomon.
- **$O(n \log n)$** -o'rta.
- **$O(n)$** - yaxshi.

Bu yerda n saralanadigan elementlar soni bo'lib, u avvaldan berilgan bo'lishi kerak.

Xotira – ba'zi bir algoritmlar saralashda qo'shimcha xotira talab etiladi.

Odatda bunday algoritmlar $O(\log n)$ xotirani talab etiladi. Ba'zi birlari saralashda qo'shimcha xotira talab etmaydi. Bunday algoritmlar o'z o'rnida (joyida) saralash deb ataladi.

Massivlarni saralash xossalari va ularning sinflari.

Turg'unlilik (stability) – turg'un saralash teng elementlarning o'rinlarini o'zaro almashtirmaydi. Agar ular bir nechta maydonlardan tashkil topgan bo'lsa, bunday xususiyatlar juda ham foydali bo'lishi mumkin.

Tabiiy xulqlilik – algoritm o'zini tabiiy holdagidek tutadi. Agar kiritiladigan ketma-ketlikdagi bu xarakteristikani xisobga olsa va yaxshi ishlasa u tabiiy xulqli deyiladi. Algoritmning asosiy xossalaridan biri unga qo'llanish sferasi(sohasi) bilan bog'liq.

Asosan ikkita turdagi saralash mavjud:

Ichki saralash. Boshqacha qilib aytganda, bunday saralash massivlarda saralashni amalga oshiradi. Bunda keltirilgan ketma – ketlik operativ xotirada joylashadi va bunda ixtiyoriy yacheykaga ruxsatli kirish mavjud. Asosan bu erda o'z joyida saralash amalga oshiriladi.

Tashqi saralash. Bu yerda fayllar ustida saralash amalga oshiriladi. Albatta, bunday saralashda vaqt ko'p ketadi, lekin o'lcham jihatdan katta ketma-ketliklarni saralash mumkin.

Saralash bo'yicha algoritmlar quyidagi sinflarga bo'linadi:

- Qo'shimcha xotiraga ehtiyoj yoki unga ehtiyojni yo'qligi.
- Solishtirish amalidan tashqariga chiquvchi ma'lumotlar tarkibi haqidagi bilimlarga ehtiyoj bor yoki unga ehtiyojni yo'qligi.

Turg'un saralash algoritmlari:

- Tanlash saralash (Selection sort) – algoritm murakkabligi $O(n^2)$.
- Ko'pikli saralash (Bubble sort) - algoritm murakkabligi $O(n^2)$.
- Aralashtirish saralashi (Sheyker, Cocktail sort, bidirectional bubble sort).

- algoritm murakkabligi $O(n^2)$.
- O‘rniga qo‘yish saralashi (Insertion sort) - algoritm murakkabligi $O(n^2)$.
- Qo‘shilish saralashi (Merge sort) - algoritm murakkabligi $O(n \log n)$.
- Ikkilik daraxti yordamida saralash (Tree sort) - algoritm murakkabligi $O(n \log n)$ qo‘shimcha $O(n)$ xotira talab etadi.
- Timsort saralashi (Timsort) - algoritm murakkabligi $O(n \log n)$ qo‘shimcha $O(n)$ xotira talab etadi.
- Sanash orqali saralash (Counting sort) - algoritm murakkabligi $O(n+k)$ qo‘shimcha $O(n)$ xotira talab etadi.
- Blokli saralash (Savatli saralash, Bucket sort) - algoritm murakkabligi $O(n)$ qo‘shimcha $O(k)$ xotira talab etadi.

Turg‘un bo‘lmagan saralash algoritmlari:

- Tanlash orqali saralash (Selection sort) algoritm murakkabligi $O(n^2)$.
- Shell saralash (Shell sort) algoritm murakkabligi $O(n \log^2 n)$.
- Tarash orqali saralash (Comb sort) algoritm murakkabligi $O(n \log n)$.
- Suzuvchi saralash (Smooth sort) algoritm murakkabligi $O(n \log n)$.
- Tez saralash (Quick sort) algoritm murakkabligi $O(n \log n)$.
- Intro sort - algoritm murakkabligi $O(n \log n)$.
- Patience sorting - algoritm murakkabligi $O(n \log n)$.
- Stooge sort - algoritm murakkabligi $O(n^{2.71})$.
- Razryadli saralash. algoritm murakkabligi $O(n+k)$. $O(k)$ qo‘shimcha xotira talab etiladi.

Amaliy bo‘lmagan saralash algoritmlari:

- Bogosort - algoritm murakkabligi $O(n \cdot n!)$.
- O‘rinlashtirish saralash - algoritm murakkabligi $O(n \cdot n!)$.
- Ma’nosiz saralash (Stupid sort) - algoritm murakkabligi $O(n^3)$.
- Bead sort - Algoritm murakkabligi $O(n)$ yoki $O(\sqrt{n})$. Maxsus apparat ta’minoti talab etiladi.
- Quymoqli saralash (Pancake sorting) - Algoritm murakkabligi $O(n)$. Maxsus apparat ta’minoti talab etiladi.

Saralash algoritmlarini quyidagi guruhlarga ajratish mumkin (6.1-rasm):



6.1-rasm. Saralash algoritmlari.

Saralash usullarini taqqoslash.

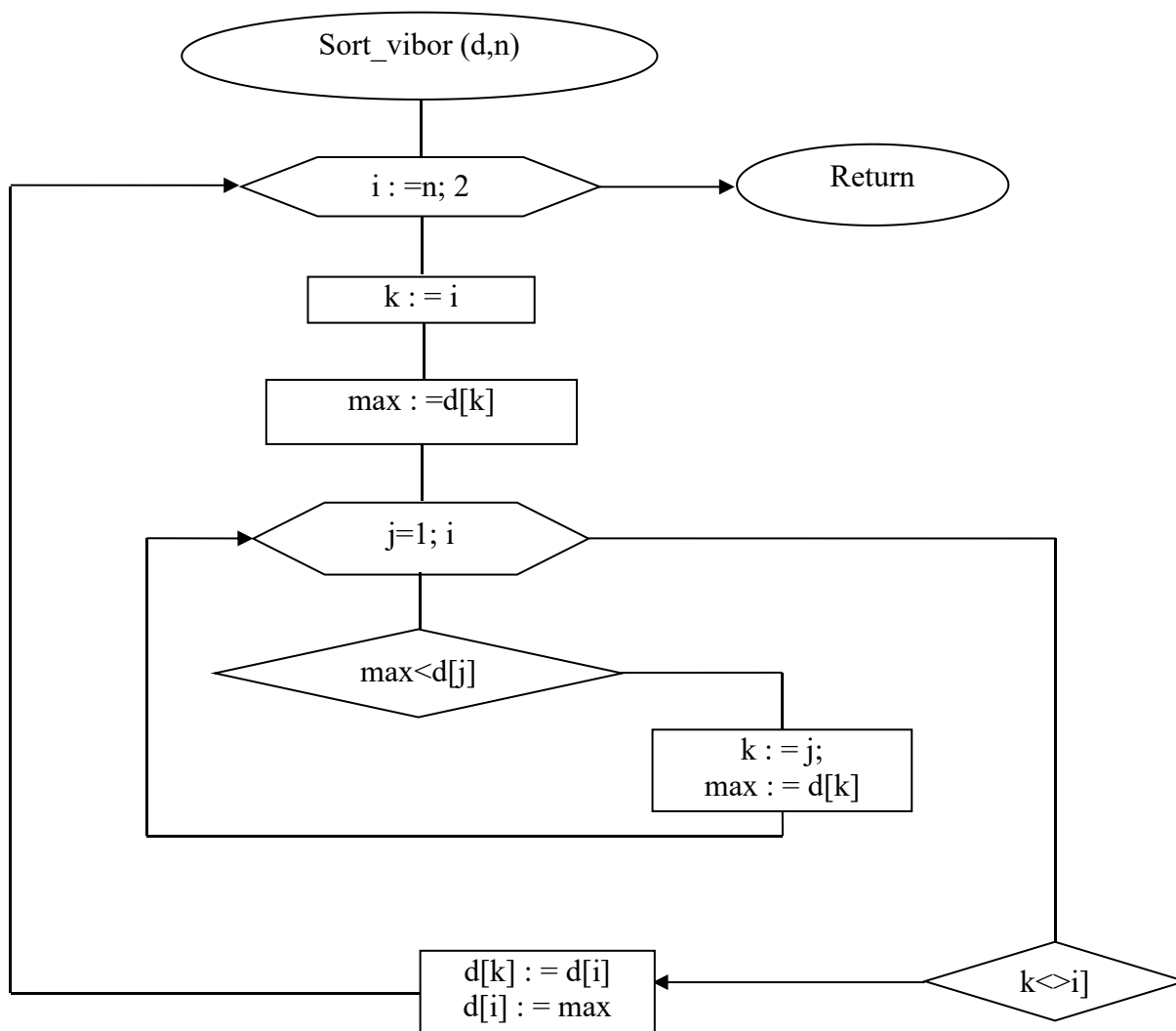
Algoritmlar	Eng yomon holda ish vaqti	O'rtacha (kutilayotgan) holda ish vaqti
Joylashtirish usulida saralash	$\theta(n^2)$	$\theta(n^2)$
Qo'shish usulida saralash	$\theta(n \lg n)$	$\theta(n \lg n)$
Piramida usulida saralash	$\theta(n \lg n)$	-

Tezkor saralash	$\theta(n^2)$	$\theta(n \lg n)$ (kutilayotgan)
Sanash usulida saralash	$\theta(k + n)$	$\theta(k + n)$
Tashlash usulida saralash	$\theta(d(n + k))$	$\theta(d(n + k))$
Cho‘ntak saralash	$\theta(n^2)$	$\theta(n)$ (o‘rtacha holda)

Eng kichik elementlari orqali tartiblash usuli. Eng kichik elementlari orqali tartiblash usuli eng sodda ko‘p qo‘llanadigan usullardan biri hisoblanadi. Faraz qilaylik, N ta elementli massiv elementlarini o‘rish tartibida tartiblash talab qilingan bo‘lsin. Uning amalga oshirish g‘oyasi quyidagicha. Dastlab massivning eng kichik elementi topiladi va birinchi element bilan o‘rin almashadi. Demak, 1-element tartiblandi. Endi qolgan elementlar orasidan eng kichigi topiladi. U ikkinchi element bilan o‘rin almashadi. Natijada ikkinchi element ham tartiblandi. Bu jarayon dastlabki N-1 ta element uchun bajariladi. Natijada massivning hamma elementlari tartiblanib qoladi. Bu g‘oyaga asoslanib, qo‘yilgan masalaning yechish algoritmini quyidagicha qurish mumkin:

1. Boshlansin
2. Kiritilsin A massiv
3. Kiritilsin n;
4. $k := 1$;
5. $\min = a(k)$; $\text{ind} = k$
6. $l = k + 1$
7. agar $a(l) < \min$ bo‘lsa, u holda $\min = a(l)$; $\text{ind} = l$
8. $l := l + 1$
9. agar $l \leq n$ bo‘lsa 7 ga o‘tilsin
10. $c := a(\text{ind})$; $a(\text{ind}) := a(k)$; $a(l) := c$;
11. Chiqarilsin $a(k)$
12. $k := k + 1$
13. agar $k \leq n - 1$ bo‘lsa, 5 ga o‘tilsin
14. Ishni tugatilsin.

Tanlash usulida saralash. Tanlash usulida saralash boshida tartibsiz ro'yhatdan eng kichik elementni tanlanashdan iborat. Shundan so'ng dastlabki ro'yhat o'zgaradi. O'zgartirilgan ro'yhat boshlang'ich ro'yhat sifatida qabul qilinadi va jarayon barcha elementlar tanlangungacha davom etadi. Tanlangan elementlar tartiblangan ro'yhatni hosil qiladi.



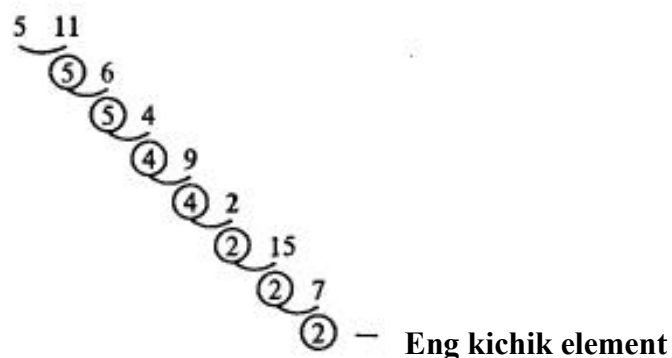
6.2-rasm. Tanlash usulida saralash blok-sxemasi

Tanlash usulida saralash eng kichik saralanmagan element aniqlanib massivning saralangan qismining oxiriga joylashtiriladi. Bu hol massivning barcha elementlari saralanib bo'lguncha davom etadi.

6.1-misol. Tanlangan elementlar tartiblangan ro'yhatni hosil qiladi. Masalan, ro'yhatdan eng kichik elementni topish talab etilsin:

{5, 11, 6, 4, 9, 2, 15, 7}.

Tanlanayogan elementlar kichik hajmda aylanaga olingan. Taqqoslanayotgan elementlar soni rasmdagi satrlar soniga mos kelishini, shuningdek, elementlarni ko'chirish soni tanlangan elementlarni o'zgartirish soniga mos kelishini ko'rish qiyin emas {5, 11, 6, 4, 9, 2, 15, 7}.



Boshlang'ich ro'yhatdan tanlangan eng kichik element unga berilgan joyga bir qancha usullar bilan joylashadi:

- Eng kichik element i - marta i - marta ($i=1, 2, \dots, n$) yangi ro'yhat joyiga ko'chirilganda, boshlang'ich ro'yhatning tanlangan element joyiga boshqa katta element joylashadi, bu bilan berilgan ro'yhat uzunligi o'zgarmaydi. Ushbu usulda o'zgartirilgan ro'yhatni boshlang'ich ro'yhat sifatida qabul qilish mumkin.

- Eng kichik element boshlang'ich ro'yhatning ($i=1, 2, \dots, n$) i chi joyiga joylashadi, i -joyning elementi esa tanlangan element joyiga joylashadi.

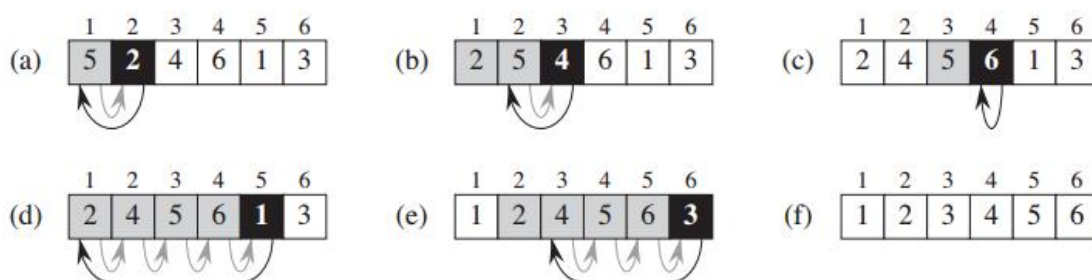
- Shu bilan birga ko'rinib turibdiki, tartiblangan elementlar keyingi saralashdan chiqariladi, shuning uchun ham har bir keyingi ro'yhatning uzunligi oldingi ro'yhatdan bitta elementga kam bo'lishi kerak.

- Tanlangan eng kichik element oldingi holda kabi berilgan ro'yhatning i chi joyiga, i chi joy keyingi eng kichik elementni yozish uchun bo'shashi uchun tanlangan elementdan ro'yhatning chap tomonda turgan qismi o'ng tomonga bitta pozitsiya bilan tanlangan element joyni to'ldirish uchun siljiydi. (Ro'yhat elementlari sikl bo'yicha siljiydi).

Tanlash usulida saralash murakkabligi $O(n^2)$ tartibda tashkil qiladi.

6.2-misol. $A = \{5, 2, 4, 6, 1, 3\}$ massivni joylashtirish usulida saralashdir. (**Insertion-sort**). Kvadratlar bilan belgilangan massiv elementlarining yuqori

qismida elementlar indeksleri va kvadratlar ichida mos elementlar qiymatlari berilgan. Bu rasmning **a-d** qismi **for** sikliga mos keladi. 1-8 qatorida for sikl iteratsiya psevdokodlari **(a)-(d)** rasm qismiga mos keladi. Har bir iteratsiyada qora kvadratlarda $A[j]$ kaliti qiymatlari tashkil topgan bo'lib, undan chapda joylashgan (psevdakodning 5-satri) kulrang kvadratlar bilan taqqoslanadi. Faqat bitta pozitsiyaga o'ng tomonga siljiydigan massivlar qiymati kulrang ko'rsatgich bilan ko'rsatilgan (6-satr), qora rangli ko'rsatgich orqali – kalitning ko'chishi ko'rsatilgan (8-satr). **(e)** qismida saralangan massivning yakuniy holati ko'rsatilgan.

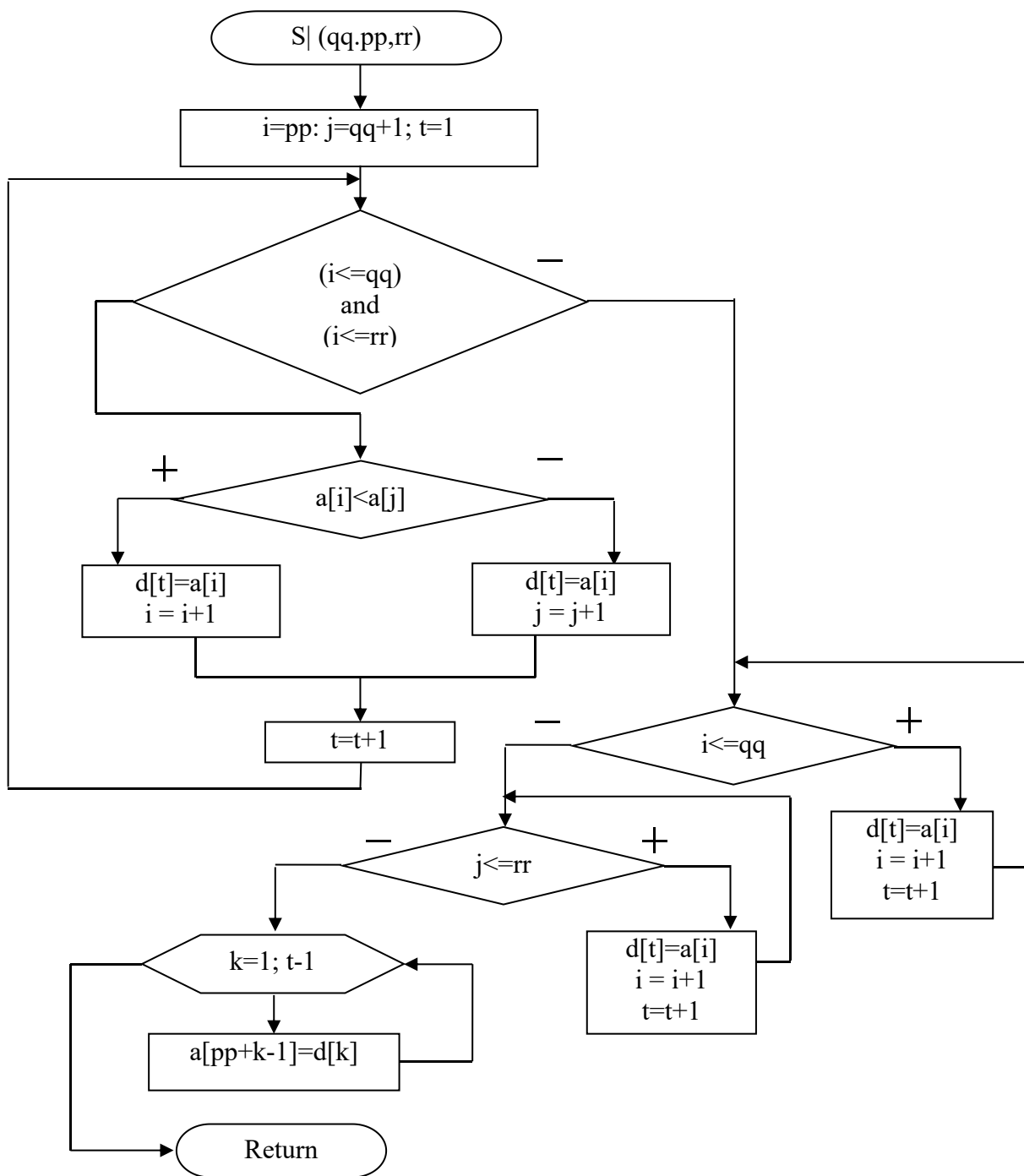


6.

3-rasm. Joylashtirish usulida saralash

Qo'shish usulida saralash. Joylashtirish usulida saralashning turlaridan biri fon Neyman usuli hisoblanadi. Ushbu masalani yechish algoritmi “fon Neyman saralash usuli” nomi bilan tanilgan yoki qo'shib saralash usuli deb nomlanadi. Ushbu usul quyidagidan iborat: avval ikkala massivning birinchi elementlari tahlil qilinadi. Eng kichik element yangi massivga yozilib boriladi. Ketma-ketlikning qolgan elementlari boshqa massiv elementlari bilan taqqoslanadi. Har bir taqqoslashdan keyin yangi massivga eng kichik element borib tushadi. Jarayon massivlarning birida elementlarning kamayishigacha davom etadi. Shunday so'ng, boshqa massivning qoldig'i yangi massivga yoziladi. Hosil qilingan yangi massiv dastlabki massiv singari shu usulda tartiblangan bo'ladi.

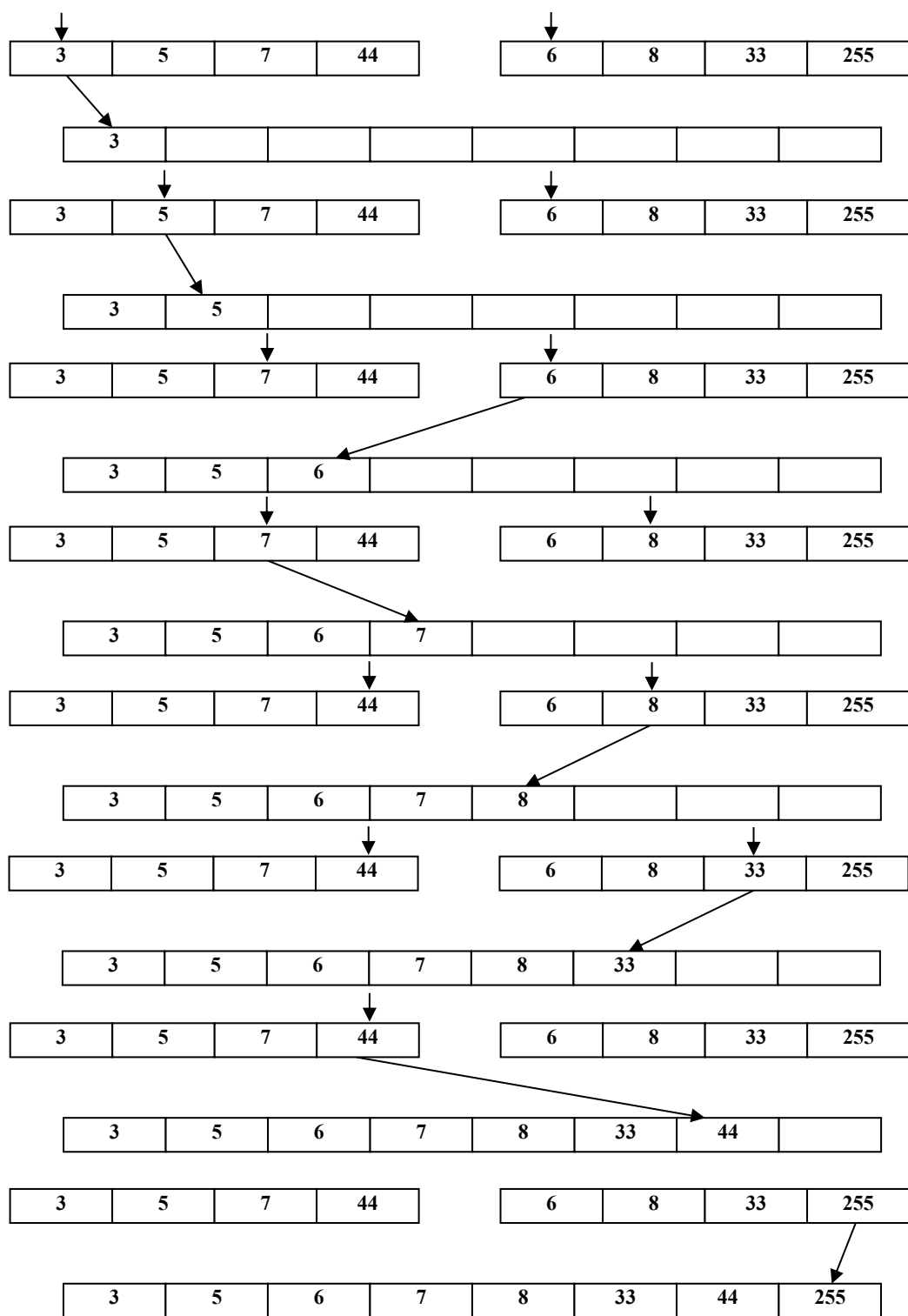
Ikkita massivning qo'shish usuli blok-sxemasi:



6.4-rasm. Qo'shish usulida saralash blok-sxemasi

Ikkita o'sish tartibida tartiblangan $p[1], p[2], \dots, p[n]$ va $q[1], q[2], \dots, q[n]$ massivlar bo'lsin va p va q massivlar qiymatlari bilan o'sish tartibida to'ldirilishi kerak bo'lgan $r[1], r[2], \dots, r[2n]$ ko'rinishidagi bo'sh massiv mavjud bo'lsin. Qo'shish uchun quyidagi amallar bajariladi: $p[1]$ va $q[1]$ lar taqqoslanadi va eng kichik qiymat $r[1]$ ga yoziladi. Ushbu qiymat $p[1]$ deb faraz qilamiz. U holda $p[2]$ bilan $q[1]$ taqqoslanadi va eng kichik qiymat $r[2]$ ga yoziladi. Ushbu qiymat $q[1]$ deb faraz qilamiz. U holda keyingi qadamda $p[2]$ va $q[2]$ lar taqqoslanadi,

jarayon biron-bir massiv chegarasiga yetib borilgunigacha davom ettiriladi. U holda boshqa massivning qoldig'i r massiv oxiriga yozib qo'yiladi.



6.5-rasm. O'sish tartibida saralash

Savol va topshiriqlar

1. Massivlarni saralash deganda nimani tushunasiz?

2. Massivlarni Saralash algoritmlarining baholanishi qanday asosiy me'zonlarga asoslanadi?

3. Massivlarni eng kichik elementlari orqali tartiblash usuli algoritmini tushuntirib bering.

4. Saralashning tanlash usulini eng kichik elementlari orqali saralash usulidan qanday farqli jihatlari bor?

5. Saralash algoritmiga misollar keltiring?

6. Saralash usullarini ayting?

7. Saralash usullarini taqqoslashni tushuntiring?

8. Tanlash usulida saralash orqali ro'yhatdan eng kichik elementni toping: **{16, 10, 5, 4, 1, 2, 15, 7}**.

9. Tanlash usulida saralash orqali ro'yhatdan eng kichik elementni toping: **{12, 10, 8, 6, 13, 21, 10, 15}**.

10. Joylashtirish usulida saralash orqali ro'yhatdan eng kichik elementni toping: **{30, 11, 73, 57, 32, 21, 79, 62}**