

## SARALASH ALGORITMLARI

Abdullayev Shaxboz Solijon o'g'li

FarDU Axborot texnologiyalari kafedrası katta o'qituvchisi, PhD

[shaxbozfardu2023@gmail.com](mailto:shaxbozfardu2023@gmail.com)

ORCID ID 0000-0001-9382-732X

O'ktamova Fotimaxon Rustamjon qizi

FarDU talabasi

<https://doi.org/10.5281/zenodo.15392121>

**Annotatsiya.** Mazkur maqola saralash algoritmlarining nazariy asoslarini chuqurroq o'rganishni istagan dasturchilar, kompyuter fanlari talabalariga, shuningdek, umumiy texnologiyalarga qiziqqan o'quvchilarga foydali manba bo'lishi mumkin. Algoritmlar bo'yicha kiritilgan tahlillar va misollar yordamida o'quvchilar ma'lumotlarni samarali tarzda tartibga solishning eng yaxshi metodlarini aniqlashda yordam oladilar. Maqola, shuningdek, algoritmlarning ishlash jarayonlaridagi optimizatsiyalarni va yuqori samaradorlikni ta'minlashga qaratilgan ilg'or usullarni ham o'z ichiga oladi. Saralash algoritmlari kompyuter fanining asosiy tushunchalaridan biri bo'lib, ma'lumotlarni tartibga solish va tizimlashtirishda keng qo'llaniladi. Ma'lumotlar bilan ishlashda samaradorlikni oshirish, tezlikni yaxshilash va xotira resurslaridan optimal foydalanish uchun turli xil saralash metodlari ishlab chiqilgan. Ushbu maqolada, eng mashhur saralash algoritmlari, jumladan, Selection Sort (Tanlab saralash), Bubble Sort (Pufakchali saralash), Insertion Sort (Joylashtirib saralash), Quick Sort (Tezkor saralash), Merge Sort (Qo'shib saralash) va Radix Sort kabi metodlar keng yoritilgan. Har bir algoritmnining ishlash prinsipi, afzalliklari, kamchiliklari, vaqt va xotira murakkabliklari tahlil qilingan. Ushbu maqola o'qituvchilar, talaba dasturchilar va kompyuter faniga qiziqadigan har bir kishi uchun saralash algoritmlarini chuqurroq tushunishga yordam beradi va ulardan samarali foydalanish yo'llarini ko'rsatadi.

**Kalit so'zlar:** saralash algoritmlari, Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Radix Sort, vaqt murakkabligi, xotira murakkabligi, algoritmlar samaradorligi, dasturlash, ma'lumotlar strukturalari, tizim optimizatsiyasi.

Saralash bu odatda ma'lumotlar elementlarining ma'lum bir atributiga asoslanib, ma'lumotlar elementlari to'plamini ko'tarilish yoki pasayish tartibida tartibga solish jarayoni sifatida qarash hisoblanadi.

**Аннотация:** данная статья может стать полезным ресурсом для программистов, студентов-информатиков, желающих углубиться в теоретические основы алгоритмов сортировки, а также для читателей, интересующихся общими технологиями. С помощью анализа и примеров, включенных в алгоритмы, учащиеся получают помощь в определении лучших методов эффективной организации данных. В статье также рассматриваются оптимизации процессов работы алгоритмов и передовые методы, направленные на обеспечение высокой эффективности. Алгоритмы сортировки являются одним из основных понятий информатики и широко используются для организации и систематизации данных. Для повышения эффективности работы с данными, повышения скорости и оптимального использования ресурсов памяти разработаны различные методы сортировки. В этой статье будут рассмотрены наиболее популярные алгоритмы сортировки, в том числе Selection sort (выборочная сортировка), Bubble.

**Ключевые слова:** алгоритмы сортировки, сортировка выбора, пузырьковая сортировка, сортировка вставки, быстрая сортировка, сортировка слияния, Радикальная сортировка, временная сложность, сложность памяти, эффективность алгоритмов, Программирование, структуры данных, оптимизация системы.

Сортировка обычно рассматривается как процесс упорядочивания набора элементов данных по возрастанию или убыванию на основе определенного атрибута элементов данных.

**Abstract.** this article can be a useful resource for programmers, computer science students who want to delve deeper into the theoretical foundations of sorting algorithms, as well as readers interested in general technology. With the analysis and examples included in the algorithms, readers will be helped to determine the best methods for efficiently organizing data. The article also includes advanced methods aimed at providing optimizations and high efficiency in the processing processes of algorithms. Sorting algorithms are one of the key concepts in Computer Science and are widely used in data regulation and systematization. Various sorting techniques have been developed to improve efficiency, improve speed, and optimally use memory resources when working with data. In this article, the most popular sorting algorithms, including Selection Sort (selective sorting), Bubble.

**Keywords:** sorting algorithms, Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Radix Sort, Time complexity, memory complexity, algorithm efficiency, Programming, Data Structures, system optimization.

*Sorting is generally considered to be the process of ordering a set of data items in ascending or descending order, based on a particular attribute of the data items.*

#### **Saralashning xususiyatlari:**

**Vaqt murakkabligi:** algoritmni ishlatish uchun qancha vaqt ketishini o'lchash. jarayonning vaqt murakkabligini aniqlash uchun saralash algoritmining eng yomon, o'rtacha va eng yaxshi ko'rsatkichlaridan foydalanish mumkin.

**Xotira murakkabligi:** saralash algoritmlari xotira murakkablikka ham ega, bu algoritmni bajarish uchun zarur bo'lgan xotira miqdori.

**Barqarorlik:** Agar saralashdan keyin teng elementlarning nisbiy tartibi saqlanib qolsa, saralash algoritmi barqaror deyiladi. Bu teng elementlarning asl tartibini saqlash kerak bo'lgan ba'zi dasturlarda muhimdir.

**Joyida saralash:** joyida saralash algoritmi-bu ma'lumotlarni saralash uchun qo'shimcha xotirani talab qilmaydigan algoritm. Bu mavjud xotira cheklangan yoki ma'lumotlarni ko'chirish mumkin bo'lmaganda muhimdir.

**Moslashuvchanlik:** adaptiv saralash algoritmi-bu ishlashni yaxshilash uchun ma'lumotlarda oldindan mavjud bo'lgan tartibdan foydalanadigan algoritm.

Biror bir ma'lumotni saralash yoki qandaydir qolipga solish juda ham muhim. Sababi, tartibsiz ma'lumotlar bilan ishlash doimo noqulayliklar keltirib chiqaradi va bunday tizim sekin va xatoliklarga moyil bo'ladi.

#### **Saralash algoritmining turlari:**

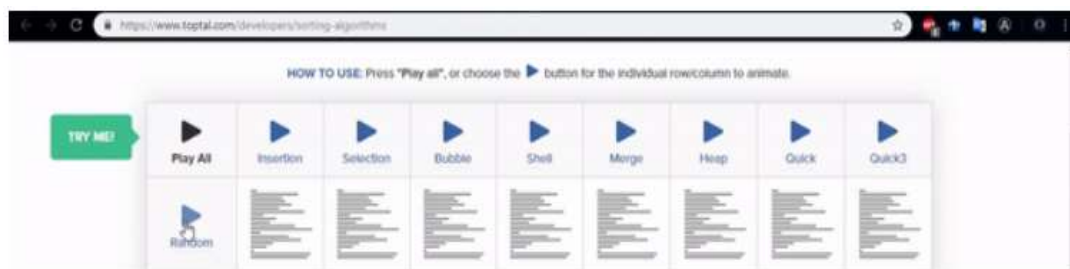
1. Selection sort (Tanlab saralash)
2. Bubble sort (Pufakchali saralash)
3. Insertion sort (Joylashtirib saralash)
4. Quick sort (Tezkor saralash)
5. Merge sort (Qo'shib saralash)
6. Radix sort

Ularning deyarli hammasi (6-sidan tashqari) ma'lumotlarni taqqoslab ko'rish orqali saralaydi va tayyor saralangan arrayni javob sifatida beradi. Birinchi 3 ta algoritm  $O(n^2)$  vaqtda ishlasa, 4-5 lari  $O(n \log n)$  vaqtda ishlaydi. Algoritmilar bir xil ishni bajarsa va ularning aksariyatining ishlash vaqti ham bir xil bo'lsa, unda ularning hammasi nimaga kerak degan haqli savol tug'iladi.

Algoritm xilma-xilligiga ikkita asosiy sabab keltirish mumkin:

Algoritmlarning ishlash vaqtlari har doim ham bir xil bo'lmaydi va ularning ishlashi qandaydir ma'lum holatlarda o'zgarib turadi. Ya'ni, umumiy holatda biror algoritmdan yomonroq ishlovchi boshqa bir algoritm, aynan, qandaydir holat uchun undan ko'ra yaxshiroq ishlashi mumkin.

Buni tushunish uchun quyidagi misolni keltiramiz. Bu yerda turli xil saralash algoritmlarining ishlashi vizual holda bir biri bilan taqqoslangan. Birinchi holatda kiruvchi ma'lumotlar, ya'ni saralanishi kerak bo'lgan ma'lumotlar ixtiyoriy turda va holatda:



Bu holatda Heap, Merge, Quick sort kabi algoritmlar o'z ishini boshidagi 3 ta algoritmdan ko'ra ancha tez yakunlayapti.

Endi esa bu misolga e'tibor bering. Endi saralanishi kerak bo'lgan ma'lumotlar to'liq bo'lmasa ham, deyarli saralangan holatda:



Bu holda Insertion sort (birinchi turgani) algoritmi yuqorida aytilgan murakkab algoritmlardan ko'ra bir necha barobar tez ishlashini ko'rishingiz mumkin.

Yuqorida faqat ikki xil holatni, ixtiyoriy va deyarli saralangan ma'lumotlarni saralanganligini ko'rish mumkin. Bunday holatlar esa ko'plab topiladi, masalan teskari saralangan, bir turdagi ma'lumotlar ko'p yoki kam bo'lgan va h.k ma'lumotlarni saralash. Har bir holat uchun ma'lum bir algoritmlar qolganlaridan ko'ra tezroq yoki sekinroq ishlab qolishi mumkin.

Ikkinchi sabab sifatida esa, albatta, saralash algoritmining xotiradan qo'shimcha joy egallashi va uning turg'unlik xususiyati inobatga olinadi.

Saralash algoritmlarida **turg'unlik** (stability) deganda, ikkita bir xil elementning ilk holatdagi bir biriga nisbatan o'rnini saralashdan keyin ham saqlab qolishiga aytiladi.

Masalan, 3 1 2 4 1 5 sonlari bor deylik, ularni saralmoqchimiz. Agar biz qo'llagan algoritm saraladan keyin **doim** birinchi 1 sonini ikkinchi 1 sonidan doim oldin joylashtirsa, bu algoritm **turg'un saralovchi algoritm** deyiladi.

Yana haqli savol tug'ilishi mumkin, "Bu narsaning kimga keragi bor, baribir natija 1 1 2 3 4 5 bo'ladiku?" degan. Albatta, bu holatda turg'unlik ahamiyati sezilmasligi mumkin. Lekin, aytaylik siz biror korxona ishchilari ma'lumotlarini ularning nomiga ko'ra saralagan paytda turg'unlik kerak bo'lib qolishi mumkin. Ya'ni, birinchi Nodirbek ma'lumotlari, ikkinchi Nodirbek ma'lumotlaridan keyin turishi kerak degan kabi.

Saralash algoritmlari ichidagi Quick Sort ko'p hollarda Merge yoki Heap sortdan tez ishlagani bilan u turg'un saralash algoritmi hisoblanmaydi (Turg'un holga keltirishning iloji bor).

**Xulosa;** Saralash algoritmlari kompyuter fanining eng muhim tushunchalaridan biri bo'lib, ular ma'lumotlarni samarali tarzda tartibga solishda va tizimlarni optimallashtirishda muhim rol o'ynaydi. Ushbu maqolada ko'rib chiqilgan **Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort** va **Radix Sort** kabi algoritmlar har biri o'zining noyob ishlash prinsiplari, samaradorligi va murakkablik darajasi bilan tanilgan. Har bir algoritmning afzalliklari va kamchiliklari turli sharoitlarda alohida o'rin tutadi va ular katta hajmdagi ma'lumotlar bilan ishlashda yoki kichik ro'yxatlar bilan ishlashda samarali bo'lishi mumkin. Shuningdek, algoritmlarning vaqt va xotira murakkabliklari, ularning qanday sharoitlarda qo'llanilishi kerakligi haqida chuqur tahlil qilindi. Saralash algoritmlarini chuqurroq o'rganish, dasturchilar, tizimlar optimizatsiyasi va ma'lumotlar strukturalari bilan ishlovchi mutaxassislar uchun zarur bo'lib, bu algoritmlardan to'g'ri foydalanish samarali tizimlar yaratishda yordam beradi. Ularning turli vaziyatlarga moslashuvchanligi va yuqori samaradorligi kelajakda innovatsion texnologiyalarni yaratishda katta ahamiyat kasb etadi.

#### **Foydalanilgan adabiyotlar:**

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed., pp. 19–26). MIT Press.
2. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed., pp. 45–52). Addison-Wesley Professional.
3. Knuth, D. E. (1998). The Art of Computer Programming, Volume 3: Sorting and Searching (2nd ed., pp. 80–87). Addison-Wesley.
4. Skiena, S. S. (2020). The Algorithm Design Manual (3rd ed., pp. 103–110). Springer.

5. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). The Design and Analysis of Computer Algorithms (pp. 65–72). Addison-Wesley.
6. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data Structures and Algorithms in Java (6th ed., pp. 201–208). Wiley.
7. Baase, S., & Van Gelder, A. (2000). Computer Algorithms: Introduction to Design and Analysis (3rd ed., pp. 123–130). Addison-Wesley.
8. Weiss, M. A. (2012). Data Structures and Algorithm Analysis in C++ (4th ed., pp. 167–174). Pearson.
9. Horowitz, E., Sahni, S., & Rajasekaran, S. (1998). Fundamentals of Computer Algorithms (2nd ed., pp. 89–96). Computer Science Press.
10. Brassard, G., & Bratley, P. (1996). Fundamentals of Algorithmics (pp. 55–62). Prentice Hall.