

**KLASSIK QIDIRUV VA SARALASH ALGORITMLARINING
PYTHON VA GNU OCTAVE MUHITLARIDA QIYOSIY TAHLILI**

Xamroyev U.N. – Navoiy davlat universiteti, p.f.f.d.(PhD), dots.v.b.

Raxmonova G.X. – Navoiy davlat universiteti, magistr

Abdusalomova M.U. – Navoiy davlat universiteti, talaba

Annotatsiya. Ushbu tadqiqotda algoritmlarning nazariy asoslari, matematik modellari, vaqt murakkabligi baholari, eksperimental natijalari hamda jadval va grafiklar asosidagi qiyosiy tahlillari keltirildi. Log–log diagrammalar orqali olingan natijalar algoritmlarning nazariy murakkabliklari bilan to'liq mos kelishi keltirildi.

Kalit so'zlar: klassik algoritmlar, qidiruv algoritmlari, saralash algoritmlari, Bubble Sort, Quick Sort, Python, GNU Octave, vaqt murakkabligi.

Abstract. This study presents the theoretical foundations of the algorithms, mathematical models, time complexity estimates, experimental results, and comparative analysis based on tables and graphs. It is shown that the results obtained through log-log diagrams fully correspond to the theoretical complexities of the algorithms.

Keywords: classical algorithms, search algorithms, sorting algorithms, Bubble Sort, Quick Sort, Python, GNU Octave, time complexity.

Kirish. Algoritmlar nazariyasi kompyuter fanining fundamental yo'nalishlaridan biri bo'lib, uning asoslari XX asr o'rtalarida shakllangan. Zamonaviy dasturiy tizimlarda algoritmlarning samaradorligi, ma'lumotlarni qayta ishlash tezligi va aniqligi muhim ahamiyatga ega. Klassik algoritmlar, ya'ni qidiruv, saralash va rekursiv algoritmlar – algoritmik fikrlashni shakllantirishda hamda murakkab hisoblash tizimlarini yaratishda muhim ahamiyat kasb etadi [1].

Qidiruv va saralash algoritmlari ma'lumotlarni qayta ishlashning eng asosiy masalalari sifatida ko'plab tadqiqotlarda o'rganilgan.

Ushbu jarayonlarning samaradorligini oshirish bevosita qo'llanilayotgan algoritmlarga bog'liq bo'lib, klassik algoritmlar ta'limiy va amaliy masalalarni yechishda asosiy vosita hisoblanadi.

Masalaning qo'yilishi. Berilgan $A = \{a_1, a_2, \dots, a_n\}$ sonli massiv uchun qidiruv va saralash algoritmlarining samaradorligini Python va GNU Octave muhitlarida bir xil eksperimental sharoitda baholash masalasini qaraymiz.

Bubble Sort algoritmi oddiy tuzilishga ega bo'lib, eng yomon holatda vaqt murakkabligi $O(n^2)$ ga teng. Quick Sort algoritmi esa "Bo'lib tashla va hukmronlik qil" strategiyasiga asoslanadi hamda o'rtacha holatda $O(n \log n)$ murakkablikni ta'minlaydi [2].

Quick Sort algoritmi uchun rekurrent tenglama: $T(n) = 2T(n/2) + O(n)$.

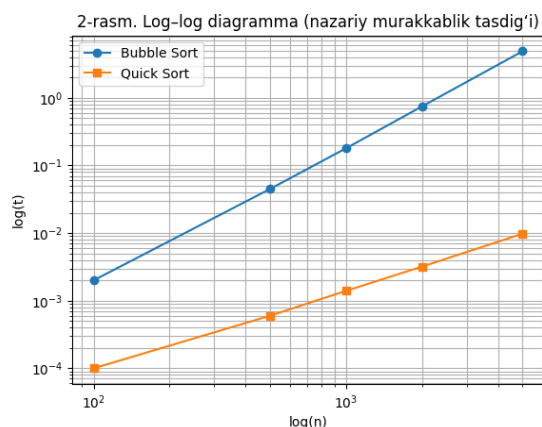
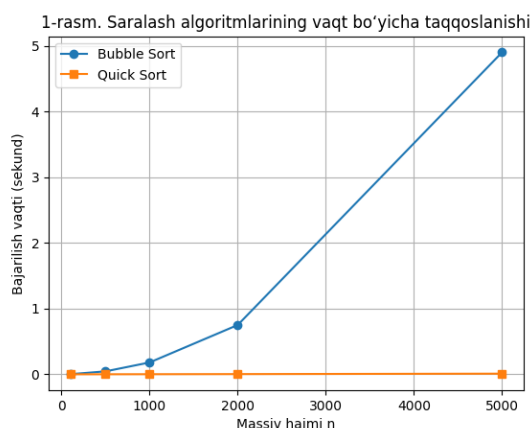
Algoritmlarning qiyosiy jadvali

Algoritm	Asosiy g'oya	Vaqt murakkabligi	Xotira sarfi	Qo'llanish sohasi
Chiziqli qidiruv (Linear Search)	Ketma-ket qidirish	$O(n)$	Past	Kichik massivlar
Binar qidiruv (Binary Search)	Bo'lish orqali qidirish	$O(\log n)$	Past	Saralangan massivlar
Pufakcha saralash (Bubble Sort)	Qo'shni elementlarni almashtirish	$O(n^2)$	Past	Ta'limiy
Tezkor saralash (Quick Sort, rekursiv)	kichik bo'laklarga bo'lish	$O(n \log n)$	O'rtacha	Amaliy tizimlar

Tadqiqotlar tasodifiy butun sonlardan iborat massivlar uchun o'tkazildi. Massiv hajmlari $n = \{100, 500, 1000, 2000, 5000\}$. Har bir tajriba bir necha marta bajarilib, o'rtacha qiymatlar olindi.

Amaliy hisoblash natijalari. 1-rasmda Bubble Sort va Quick Sort algoritmlarining bajarilish vaqti massiv hajmiga bog'liqligi ko'rsatilgan. Grafikdan ko'rinadiki, Bubble Sort algoritmi uchun bajarilish vaqti kvadratik tezlikda ortadi, Quick Sort esa katta hajmli ma'lumotlar uchun ancha samarali ishlaydi.

2-rasmda log-log koordinatalarda qurilgan diagramma keltirilgan bo'lib, unda Bubble Sort algoritmi uchun o'sish darajasi taxminan 2 ga teng, Quick Sort algoritmi esa $n \log n$ xususiyatini namoyon etadi.



Xulosa. Klassik qidiruv va saralash algoritmlarining Python va GNU Octave muhitlarida bir xil eksperimental sharoitda kengaytirilgan qiyosiy tahlili amalga oshirilgan. O'tkazilgan tadqiqotlar natijalari klassik algoritmlarning nazariy vaqt murakkabligi baholari amaliy eksperimental natijalar bilan to'liq mos kelishini tasdiqladi. Python va GNU Octave muhitlari algoritmik tadqiqotlar va ta'lim

jarayoni uchun samarali platformalar ekanligi aniqlandi, shuningdek, Python umumiy dasturiy ta'minot va katta hajmli hisoblashlar uchun qulay bo'lsa, GNU Octave matematik modellashtirish va ilmiy bayon uchun soddaligi bilan ajralib turadi.

Foydalanilgan adabiyotlar

1. Кнут Д.Э. Искусство программирования. Том 1–3. – М.: Вильямс, 2015.
2. Кормен Т.Х., Лейзерсон Ч.Э., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. – М.: Вильямс, 2022.
3. Седжвик Р., Уэйн К. Алгоритмы. – СПб.: Питер, 2019. – 992 с.
4. Skiena S. S. The Algorithm Design Manual. – New York: Springer, 2020. – 730 p.
5. Kleinberg J., Tardos É. Algorithm Design. – Boston: Pearson, 2018. – 864 p.
6. Gaynazarov S.M., Saidov D.Y. Algoritmlar va berilganlar tuzilmasi. O'quv qo'llanma. –T.: “Ma’rifat”, 2023. 254 bet.
7. Xamroyeva D.N., Djurayev D.D., Xamroyev U.N. Algoritmlash asoslari Darslik. Navoiy: “Navoiy” nashriyoti, 2024. – 144 b.
8. Yodgorov G'R., Xamroyeva D.N., Xamroyev U.N. Algoritmlar fanidan “Saralash usullari” mavzusini o'qitishda multimediali o'quv tizimining imkoniyatlari. “Fizika-matematika va informatika”, Respublika ilmiy jurnali. ISSN 2091-5586.– Toshkent, 2018. – № 2. – B. 26-33.
9. Aho A. V., Hopcroft J. E., Ullman J. D. Data Structures and Algorithms. – Boston: Addison-Wesley, 2014.
10. Press W. H. et al. Numerical Recipes. – Cambridge: Cambridge University Press, 2016.
11. Goodrich M. T. et al. Data Structures and Algorithms in Python. – Wiley, 2016.