KATTA HAJMDAGI MA'LUMOTLARNI QAYTA ISHLASHDA QIDIRUV VA SARALASH ALGORITMLARINING VAQT VA XOTIRA MURAKKABLIGI (TIME AND SPACE COMPLEXITY) QIYOSIY TAHLILI

Ismoilov Sirojiddin Rasuljon o'g'li
Farg‘ona davlat texnika universiteti,
Dasturiy injiniring yo‘nalishi 4-bosqich talabasi
sirojiddin.ismoilov7181@gmail.com

Annotatsiya. Mazkur maqolada katta hajmdagi ma'lumotlarni qayta ishlashda qidiruv va saralash algoritmlarining vaqt va xotira murakkabligi qiyosiy tahlil qilinadi. Tadqiqotda linear search, binary search, bubble sort, insertion sort, selection sort, merge sort, quick sort va heap sort algoritmlarining samaradorligi Big-O notatsiyasi asosida baholandi. Turli hajmdagi ma'lumotlar uchun hisoblash natijalari solishtirilib, katta hajmli ma'lumotlar bilan ishlashda O(n log n) murakkablikka ega algoritmlar samaraliroq ekani aniqlandi.

Kalit so‘zlar: Katta hajmdagi ma'lumotlar, algoritm, qidiruv algoritmlari, saralash algoritmlari, vaqt murakkabligi, xotira murakkabligi, Big-O notatsiyasi, linear search, binary search, bubble sort, insertion sort, selection sort, merge sort, quick sort, heap sort, algoritmlar tahlili, samaradorlik, ma'lumotlarni qayta ishlash.

Kirish

Hozirgi kunda raqamli texnologiyalar rivojlanishi natijasida katta hajmdagi ma'lumotlarni qayta ishlash masalasi dolzarb ahamiyat kasb etmoqda. Turli axborot tizimlari, ma'lumotlar bazalari, sun'iy intellekt, mashinali o‘qitish, bank tizimlari hamda internet xizmatlarida har kuni juda katta hajmdagi ma'lumotlar hosil bo‘lmoqda. Ushbu ma'lumotlarni samarali qayta ishlash, ularni tartiblash va tezkor qidirish zamonaviy dasturiy tizimlar uchun muhim talab hisoblanadi. Ma'lumotlar hajmi ortib borishi bilan algoritmlarning ishlash tezligi va xotira sarfi muhim omilga aylanadi.

Qidiruv va saralash algoritmlari ma'lumotlarni qayta ishlashning asosiy bosqichlarini tashkil etadi. Qidiruv algoritmlari berilgan elementni massiv yoki ma'lumotlar to‘plamidan topish uchun qo‘llanilsa, saralash algoritmlari ma'lumotlarni

ma'lum tartibga keltirish uchun ishlatiladi. Amaliy dasturlashda linear search, binary search, bubble sort, selection sort, insertion sort, merge sort, quick sort va heap sort kabi algoritmlar keng qo‘llaniladi. Ushbu algoritmlarning har biri turli vaqt murakkabligi va xotira sarfiga ega bo‘lib, ularni to‘g‘ri tanlash tizim samaradorligiga bevosita ta'sir ko‘rsatadi.

Algoritmlar samaradorligini baholashda vaqt murakkabligi (Time Complexity) va xotira murakkabligi (Space Complexity) asosiy mezon hisoblanadi. Vaqt murakkabligi kiruvchi ma'lumot hajmi ortgan sari algoritm bajarilish vaqti qanday o‘zgarishini ifodalaydi. Xotira murakkabligi esa algoritm ishlashi davomida talab qilinadigan qo‘shimcha xotira hajmini aniqlaydi. Katta hajmdagi ma'lumotlar bilan ishlashda noto‘g‘ri tanlangan algoritm tizim ishlash tezligini sekinlashtirishi yoki ortiqcha xotira sarfiga olib kelishi mumkin.

Mazkur maqolada qidiruv va saralash algoritmlarining vaqt hamda xotira murakkabligi qiyosiy tahlil qilinadi. Tadqiqotda turli algoritmlar Big-O notatsiyasi asosida baholanib, ularning samaradorligi turli hajmdagi ma'lumotlar uchun solishtiriladi. Shuningdek, katta hajmdagi ma'lumotlarni qayta ishlashda qaysi algoritmlardan foydalanish maqsadga muvofiq ekani aniqlanadi. Olingan natijalar algoritm tanlash jarayonini optimallashtirish hamda dasturiy tizimlar samaradorligini oshirishga xizmat qiladi.

Adabiyotlar tahlili. Katta hajmdagi ma'lumotlarni qayta ishlashda qidiruv va saralash algoritmlarining vaqt va xotira murakkabligini o‘rganish masalasi ko‘plab olimlar tomonidan tadqiq qilingan. Ushbu yo‘nalishda algoritmlarning nazariy asoslari, samaradorligi hamda amaliy qo‘llanilishi bo‘yicha turli ilmiy ishlanmalar mavjud.

Amerikalik olim Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest va Clifford Stein tomonidan yozilgan **“Introduction to Algorithms”** asarida qidiruv va saralash algoritmlarining vaqt murakkabligi Big-O notatsiyasi asosida batafsil tahlil qilingan. Ushbu kitobda insertion sort, merge sort, heap sort va quick sort algoritmlarining eng yaxshi, o‘rtacha va eng yomon holatlari matematik jihatdan asoslab berilgan hamda katta hajmdagi ma'lumotlarda merge sort algoritmining

barqarorligi, quick sort algoritmining esa amaliy tezkorligi yuqori ekanligi ko‘rsatib berilgan.

Algoritmlar tahlili bo‘yicha yana bir muhim manba Donald E. Knuth ning **“The Art of Computer Programming, Volume 3: Sorting and Searching”** nomli fundamental asari hisoblanadi. Mazkur kitobda saralash algoritmlarining murakkabligi, ularning taqqoslashlar soni, xotira sarfi va ishlash tezligi chuqur matematik modellar orqali o‘rganilgan. Muallif tomonidan quick sort, merge sort va heap sort algoritmlarining katta hajmdagi ma'lumotlarda samaradorligi yuqori ekanligi ilmiy jihatdan asoslangan.

Robert Sedgewick va Kevin Wayne tomonidan yozilgan **“Algorithms”** nomli kitobda zamonaviy qidiruv va saralash algoritmlarining amaliy samaradorligi taqqoslangan. Ushbu manbada binary search algoritmining vaqt murakkabligi O(log n) ekani, linear search algoritmining esa O(n) murakkablikka egaligi ta'kidlangan. Shuningdek, merge sort algoritmi katta hajmdagi ma'lumotlarni tashqi xotirada saralash uchun qulay ekani ko‘rsatib o‘tilgan.

Ma'lumotlar tuzilmalari va algoritmlar samaradorligini o‘rganishda Mark Allen Weiss ning **“Data Structures and Algorithm Analysis in C++”** asari muhim o‘rin tutadi. Ushbu kitobda saralash algoritmlarining xotira murakkabligi ham tahlil qilinib, quick sort algoritmining o‘rtacha O(n log n) vaqt murakkabligiga ega bo‘lishi, merge sort algoritmining esa qo‘shimcha O(n) xotira talab qilishi ko‘rsatib berilgan.

Steven S. Skiena tomonidan yozilgan **“The Algorithm Design Manual”** kitobida algoritm tanlash muammosi katta hajmdagi ma'lumotlar bilan ishlash nuqtai nazaridan tahlil qilingan. Muallif saralash algoritmlarini amaliy samaradorlik bo‘yicha solishtirib, quick sort algoritmi ko‘pchilik real tizimlarda eng tezkor ishlashini, heap sort algoritmi esa xotira jihatidan samaraliroq ekanini ta'kidlaydi.

Qidiruv algoritmlarining samaradorligi masalasi Jon Bentley ning **“Programming Pearls”** kitobida ham ko‘rib chiqilgan. Muallif binar qidiruv algoritmini optimallashtirish muammolarini tahlil qilib, katta hajmdagi ma'lumotlar uchun logarifmik murakkablikka ega algoritmlar eng samarali yechim ekanini ko‘rsatadi.

Shuningdek, Alfred V. Aho, John E. Hopcroft va Jeffrey D. Ullman tomonidan yozilgan **“Data Structures and Algorithms”** asarida algoritmlarning nazariy murakkabligi bilan bir qatorda ularning xotira sarfi ham tahlil qilingan. Ushbu manbada katta hajmdagi ma'lumotlarni qayta ishlashda O(n log n) murakkablikka ega saralash algoritmlari optimal yechim hisoblanishi ko‘rsatib o‘tilgan.

Yuqoridagi adabiyotlar tahlili shuni ko‘rsatadiki, qidiruv va saralash algoritmlarining vaqt hamda xotira murakkabligini qiyosiy o‘rganish katta hajmdagi ma'lumotlarni samarali qayta ishlashda muhim ahamiyatga ega. Mazkur maqolada ushbu ilmiy manbalar asosida qidiruv va saralash algoritmlarining qiyosiy tahlili amalga oshiriladi.

Tadqiqot metodologiyasi

Mazkur tadqiqotda qidiruv va saralash algoritmlarining vaqt (Time Complexity) hamda xotira (Space Complexity) murakkabligi nazariy tahlil va qiyosiy solishtirish usuli asosida baholandi. Algoritmlar samaradorligi Big-O notatsiyasi yordamida aniqlanib, turli hajmdagi ma'lumotlar uchun ($n = 10^3, 10^4, 10^5, 10^6$) qiyosiy tahlil qilindi. Natijalar umumlashtirilib, matematik ifodalar asosida jadval ko‘rinishida taqdim etildi.

1-jadval. Qidiruv algoritmlarining matematik qiyosiy tahlili

Algoritm	Matematik ifoda T(n)	Best	Average	Worst	Space S(n)
Linear Search	T(n)=n	O(1)	O(n)	O(n)	O(1)
Binary Search	T(n)=log n	O(1)	O(log n)	O(log n)	O(1)

2-jadval. Saralash algoritmlarining matematik qiyosiy tahlili

Algoritm	Matematik ifoda T(n)	Best	Average	Worst	Space S(n)
Bubble Sort	$T(n)=n^2$	O(n)	$O(n^2)$	$O(n^2)$	O(1)
Selection Sort	$T(n)=n^2$	$O(n^2)$	$O(n^2)$	$O(n^2)$	O(1)

Insertion Sort	$T(n)=n^2$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Merge Sort	$T(n)=n \log n$	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Quick Sort	$T(n)=n \log n$	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$
Heap Sort	$T(n)=n \log n$	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$

Baholash mezoni:

$T_1(n) < T_2(n) \rightarrow$ algoritm samaraliroq

$S_1(n) < S_2(n) \rightarrow$ xotira tejamkorroq

Tadqiqot natijalari. Tadqiqot davomida qidiruv va saralash algoritmlarining bajarilish vaqti turli hajmdagi ma'lumotlar uchun nazariy hisoblash asosida baholandi. Hisoblashlar algoritmlarning vaqt murakkabligi formulalari asosida $n = 10^3, 10^4, 10^5, 10^6$ qiymatlar uchun amalga oshirildi.

Vaqt baholash formulasi:

$T(n) \approx f(n)$

Logarifmik qiymatlar:

$\log_2(10^3)=10$

$\log_2(10^4)=13$

$\log_2(10^5)=17$

$\log_2(10^6)=20$

3-jadval. Qidiruv algoritmlarining hisoblangan natijalari

n	Linear Search $T(n)=n$	Binary Search $T(n)=\log n$	Farq
10^3	1000	10	100x tez
10^4	10000	13	769x tez
10^5	100000	17	5882x tez
10^6	1000000	20	50000x tez

Natijadan ko‘rinadiki, katta hajmdagi ma'lumotlarda binary search linear search ga nisbatan keskin tez ishlaydi.

4-jadval. Saralash algoritmlarini solishtirma jadvali

Algoritm	Formula	10^3	10^4	10^5	10^6
Bubble Sort	n^2	1e6	1e8	1e10	1e12
Selection Sort	n^2	1e6	1e8	1e10	1e12
Insertion Sort	n^2	1e6	1e8	1e10	1e12
Merge Sort	n log n	1e4	1.3e5	1.7e6	2e7
Quick Sort	n log n	1e4	1.3e5	1.7e6	2e7
Heap Sort	n log n	1e4	1.3e5	1.7e6	2e7

5-jadval. Xotira murakkabligi solishtirma jadvali

Algoritm	Space	Qo‘shimcha xotira
Bubble Sort	O(1)	minimal
Selection Sort	O(1)	minimal
Insertion Sort	O(1)	minimal
Heap Sort	O(1)	minimal
Quick Sort	O(log n)	rekursiv
Merge Sort	O(n)	katta

6-jadval. Umumiy qiyosiy natija

Algoritm	Tezlik	Xotira	Katta hajm uchun
Linear Search	sekin	yaxshi	yo‘q
Binary Search	juda tez	yaxshi	ha
Bubble Sort	juda sekin	yaxshi	yo‘q
Selection Sort	juda sekin	yaxshi	yo‘q
Insertion Sort	sekin	yaxshi	yo‘q
Merge Sort	tez	yomon	ha
Quick Sort	juda tez	yaxshi	eng yaxshi
Heap Sort	tez	juda yaxshi	ha

Natijalar muhokamasi. O‘tkazilgan tadqiqot natijalari qidiruv va saralash algoritmlarining samaradorligi ma'lumotlar hajmiga bevosita bog‘liq ekanligini ko‘rsatdi. Hisoblangan jadval natijalariga ko‘ra, chiziqli qidiruv algoritmi T(n)=n murakkablikka ega bo‘lib, ma'lumotlar soni ortishi bilan bajarilish vaqti chiziqli ravishda oshadi. Binar qidiruv algoritmi esa T(n)=log n murakkablikka ega bo‘lgani sababli katta hajmdagi ma'lumotlar uchun ancha samarali ishlashi aniqlandi. Ayniqsa $n = 10^6$ bo‘lgan holatda chiziqli qidiruv 1 000 000 ta taqqoslash talab qilsa, binar qidiruv atigi 20 ta taqqoslash bilan natijani beradi. Bu esa qidiruv algoritmlarida logarifmik murakkablikning ustunligini tasdiqlaydi.

Saralash algoritmlari tahlili shuni ko‘rsatdiki, kvadratik murakkablikka ega algoritmlar (bubble sort, selection sort, insertion sort) katta hajmdagi ma'lumotlar uchun samarasiz hisoblanadi. Hisoblash natijalariga ko‘ra $n = 10^6$ bo‘lgan holatda ushbu algoritmlar uchun amallar soni $n^2 = 10^{12}$ ga teng bo‘ladi. Bu esa amaliy tizimlarda juda katta vaqt talab qiladi. Shu sababli ushbu algoritmlar faqat kichik hajmdagi ma'lumotlar uchun qo‘llanishi maqsadga muvofiq.

O(n log n) murakkablikka ega merge sort, quick sort va heap sort algoritmlari esa katta hajmdagi ma'lumotlarda yuqori samaradorlikni ko‘rsatdi. Hisoblangan natijalarga ko‘ra $n = 10^6$ holatda ushbu algoritmlar uchun amallar soni taxminan 2×10^7 ni tashkil qiladi. Bu kvadratik algoritmlarga nisbatan minglab marta tez ishlashini anglatadi. Shu sababli katta hajmdagi ma'lumotlarni qayta ishlashda aynan shu algoritmlar samarali hisoblanadi.

Xotira murakkabligi bo‘yicha tahlil shuni ko‘rsatdiki, merge sort algoritmi O(n) qo‘shimcha xotira talab qiladi. Bu katta hajmdagi ma'lumotlarda muammo tug‘dirishi mumkin. Quick sort algoritmi esa o‘rtacha holatda O(log n) xotira ishlatadi, ammo eng yomon holatda O(n) ga yetishi mumkin. Heap sort algoritmi esa O(1) xotira bilan ishlaydi va shu sababli xotira cheklangan tizimlar uchun eng maqbul variant hisoblanadi.

Tadqiqot natijalari shuni ko‘rsatdiki, algoritm tanlashda faqat vaqt murakkabligi emas, balki xotira sarfi ham muhim omil hisoblanadi. Masalan, merge sort tez ishlaydi, lekin qo‘shimcha xotira talab qiladi. Heap sort esa biroz sekinroq bo‘lsa ham xotira

jihatidan samaraliroq. Quick sort esa o‘rtacha holatda eng optimal muvozanatni ta'minlaydi.

Umuman olganda, tadqiqot natijalari quyidagilarni ko‘rsatdi: kichik hajmdagi ma'lumotlar uchun oddiy saralash algoritmlari yetarli bo‘ladi, o‘rta hajmdagi ma'lumotlar uchun insertion sort ba'zi holatlarda samarali bo‘lishi mumkin, katta hajmdagi ma'lumotlar uchun esa quick sort, merge sort va heap sort algoritmlaridan foydalanish maqsadga muvofiq hisoblanadi. Qidiruv jarayonida esa saralangan massivlarda binar qidiruv algoritmi eng optimal yechim sifatida tavsiya etiladi.

Foydalanilgan adabiyotlar

1. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms. — MIT Press, 2009. — 1312 p.
2. Knuth D.E. The Art of Computer Programming. Volume 3: Sorting and Searching. — Addison-Wesley, 1998. — 780 p.
3. Sedgewick R., Wayne K. Algorithms (4th Edition). — Addison-Wesley Professional, 2011. — 992 p.
4. Weiss M.A. Data Structures and Algorithm Analysis in C++ (4th Edition). — Pearson, 2013. — 640 p.
5. Skiena S.S. The Algorithm Design Manual (2nd Edition). — Springer, 2008. — 730 p.
6. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. — Addison-Wesley, 1983. — 427 p.
7. Bentley J. Programming Pearls (2nd Edition). — Addison-Wesley, 2000. — 256 p.
8. Goodrich M.T., Tamassia R., Goldwasser M.H. Data Structures and Algorithms in Python. — Wiley, 2013. — 768 p.
9. Kleinberg J., Tardos É. Algorithm Design. — Pearson, 2006. — 838 p.
10. Horowitz E., Sahni S., Anderson-Freed S. Fundamentals of Data Structures in C. — Silicon Press, 2008. — 576 p.
11. Brass P. Advanced Data Structures. — Cambridge University Press, 2008. — 432 p.
12. Lafore R. Data Structures and Algorithms in Java (2nd Edition). — Sams Publishing, 2002. — 800 p.