

PIGEONHOLE SORT - TEZKOR SARALASH ALGORITMI

Onarkulov Maksadjon Karimberdiyevich

*Farg'ona davlat univarsiteti Amaliy matematika va
informatika kafedrası dotsenti (PhD)*

maxmaqsad@gmail.com

Mashrabova Charosxon Qaxramonjon qizi

Farg'ona davlat universiteti 2-bosqich talabasi

charosxonmashrabova@gmail.com

Annotatsiya:

Pigeonhole Sort (Kabutar uyasi saralash) algoritmi maxsus sharoitlarda juda tez ishlaydigan saralash algoritmlaridan biridir. U 1956-yilda ishlab chiqilgan bo'lib, massiv elementlarining qiymatlari ma'lum va cheklangan diapazonda bo'lganda qo'llaniladi. Algoritm har bir mumkin bo'lgan qiymat uchun alohida "kabutar uyasi" (bucket) yaratadi va elementlarni mos uyalarga joylaydi, so'ngra ketma-ket o'qiydi. Pigeonhole Sort $O(n + k)$ vaqt murakkabligiga ega bo'lib, bu yerda n - elementlar soni, k - qiymatlar diapazoni. Bu xususiyat uni ayrim hollarda eng tez saralash algoritmiga aylantiradi. Algoritm sonli ma'lumotlarni saralashda, statistik tahlillarda va qiymatlar diapazoni cheklangan bo'lgan holatlarda keng qo'llaniladi. Ushbu maqolada Pigeonhole Sort algoritmining ishlash mexanizmi, afzalliklari, cheklovlari va amaliy qo'llanish sohalari batafsil yoritiladi.

Kalit so'zlar: *Pigeonhole Sort, saralash algoritmi, bucket sort, vaqt murakkabligi, $O(n+k)$, taqsimlash usuli, sonli saralash, cheklangan diapazon, xotira samaradorligi, algoritm tahlili, ma'lumotlar strukturasi, statistik saralash*

Annotation:

Pigeonhole Sort is one of the fastest sorting algorithms under specific conditions. Developed in 1956, it is used when array elements have known and limited range values. The algorithm creates separate "pigeonholes" (buckets) for each possible value and places elements into corresponding holes, then reads them sequentially. Pigeonhole Sort has $O(n + k)$ time complexity, where n is the





number of elements and k is the range of values. This characteristic makes it the fastest sorting algorithm in certain cases. The algorithm is widely used in numerical data sorting, statistical analysis, and situations where value ranges are limited. This paper provides a detailed examination of Pigeonhole Sort algorithm mechanism, advantages, limitations, and practical applications.

Keywords: Pigeonhole Sort, sorting algorithm, bucket sort, time complexity, $O(n+k)$, distribution method, numerical sorting, limited range, memory efficiency, algorithm analysis, data structures, statistical sorting

Аннотация:

Сортировка "Голубиные гнёзда" (Pigeonhole Sort) является одним из самых быстрых алгоритмов сортировки при определённых условиях. Разработанный в 1956 году, он используется когда элементы массива имеют известный и ограниченный диапазон значений. Алгоритм создаёт отдельные "голубиные гнёзда" (корзины) для каждого возможного значения и размещает элементы в соответствующие гнёзда, затем последовательно считывает их. Pigeonhole Sort имеет временную сложность $O(n + k)$, где n - количество элементов, k - диапазон значений. Эта характеристика делает его самым быстрым алгоритмом сортировки в определённых случаях. Алгоритм широко используется в сортировке числовых данных, статистическом анализе и ситуациях где диапазон значений ограничен. В данной статье подробно рассматриваются механизм работы алгоритма Pigeonhole Sort, преимущества, ограничения и практические применения.

Ключевые слова: Pigeonhole Sort, алгоритм сортировки, корзинная сортировка, временная сложность, $O(n+k)$, метод распределения, числовая сортировка, ограниченный диапазон, эффективность памяти, анализ алгоритма, структуры данных, статистическая сортировка

Saralash algoritmlari kompyuter fanining asosiy tushunchalaridan biri bo'lib, ma'lumotlarni tartibli holatga keltirish zamonaviy dasturlashning muhim



qismidir. Turli saralash algoritmlari turli sharoitlarda har xil samaradorlik ko'rsatadi. Umuman qabul qilingan solishtirishga asoslangan saralash algoritmlarining nazariy pastki chegarasi $O(n \log n)$ hisoblanadi, ammo maxsus sharoitlarda bundan tezroq ishlaydigan algoritmlar mavjud. Pigeonhole Sort aynan shunday maxsus algoritmlardan biri bo'lib, ma'lum sharoitlarda barcha boshqa saralash algoritmlaridan tezroq ishlaydi.

Pigeonhole Sort algoritmi 1956-yilda ishlab chiqilgan bo'lib, uning nomi matematikaning mashhur "Kabutar uyasi printsipi" (Pigeonhole Principle) dan olingan. Bu printsip shuni ta'kidlaydiki, agar n ta kabutar $n-1$ ta uyaga joylashtirilsa, kamida bitta uyada ikkita yoki undan ko'p kabutar bo'ladi. Saralash algoritmida bu printsip teskari yo'nalishda qo'llaniladi - har bir mumkin bo'lgan qiymat uchun alohida "uya" yaratiladi va elementlar mos uyalarga taqsimlanadi.

Bu algoritm ayniqsa **cheklangan diapazonli sonlar** bilan ishlaganda o'zining ajoyib samaradorligini namoyon etadi. Masalan, 0 dan 100 gacha bo'lgan baholar, 1 dan 12 gacha bo'lgan oylar, yoki ASCII kodlar kabi ma'lumotlarni saralashda Pigeonhole Sort boshqa barcha algoritmlarga nisbatan ancha tez ishlaydi. Algoritmning asosiy g'oyasi juda sodda: har bir mumkin bo'lgan qiymat uchun alohida joy ajratish, elementlarni mos joylarga taqsimlash va oxirda ketma-ket o'qish. Pigeonhole Sort ning muhim xususiyati shundaki, u **taqsimlash usuli** (distribution sort) guruhiga kiradi va solishtirishga asoslanmagan. Bu xususiyat unga $O(n + k)$ vaqt murakkabligini beradi, bu yerda n - elementlar soni, k - qiymatlar diapazoni. Agar k nisbatan kichik bo'lsa, algoritm amalda chiziqli vaqtda ishlaydi va bu uni eng tez saralash algoritmiga aylantiradi. Zamonaviy dasturlashda Pigeonhole Sort turli sohalarda qo'llaniladi: statistik ma'lumotlarni qayta ishlash, o'yin dasturlash (ball jadvallarini saralash), ma'lumotlar bazalarida indekslar yaratish va real vaqtda ishlashi kerak bo'lgan tizimlarda. Uning soddaligi va yuqori samaradorligi uni kichik diapazonli ma'lumotlar bilan ishlaydigan dasturchilar uchun juda qimmatli qiladi.

Pigeonhole Sort algoritmining ishlash printsipi



Pigeonhole Sort algoritmining ishlash mexanizmi uning soddaligi va samaradorligi bilan ajralib turadi. Algoritm asosan uchta asosiy bosqichdan iborat: **tayyorgarlik bosqichi**, **taqsimlash bosqichi** va **yig'ish bosqichi**. Har bir bosqich o'zining aniq vazifasiga ega va umumiy samaradorlikka hissa qo'shadi.

Tayyorgarlik bosqichida algoritm birinchi navbatda kiritilgan massivdagi elementlarning minimal va maksimal qiymatlarini topadi. Bu jarayon $O(n)$ vaqt talab qiladi va diapazonni aniqlash uchun zarur. Masalan, [8, 3, 2, 7, 4, 6, 8] massivi uchun $\min = 2$, $\max = 8$ bo'ladi. So'ngra diapazon hajmi hisoblanadi: $k = \max - \min + 1$. Bizning misolda $k = 8 - 2 + 1 = 7$ bo'ladi. Ushbu diapazon asosida k o'lchamli yordamchi massiv (pigeonhole massivi) yaratiladi.

Taqsimlash bosqichida asl massivning har bir elementi mos "kabutar uyasiga" joylashtiriladi. Element qiymati va minimal qiymat orasidagi farq indeks sifatida ishlatiladi. Masalan, qiymati 8 bo'lgan element $\text{pigeonhole}[8-2] = \text{pigeonhole}[6]$ pozitsiyasiga joylashadi. Agar bir xil qiymatli elementlar mavjud bo'lsa, ular bir xil uyaga joylashadi va ro'yxat shaklida saqlanadi. Bu bosqich ham $O(n)$ vaqt talab qiladi.

Yig'ish bosqichida pigeonhole massivi ketma-ket ko'rib chiqiladi va bo'sh bo'lmagan uyalardagi elementlar asl massivga qaytariladi. Bu jarayon ham $O(n)$ vaqt oladi, chunki jami n ta element qayta yoziladi. Natijada asl massiv to'liq saralangan holatga keladi.

Algoritmnining **psevdokodi** quyidagicha:

1. $\min_val = \text{massivdagi minimal qiymat}$
2. $\max_val = \text{massivdagi maksimal qiymat}$
3. $k = \max_val - \min_val + 1$
4. $\text{pigeonhole}[] = k \text{ o'lchamli bo'sh massiv}$
5. Har bir element uchun:
 - $\text{pigeonhole}[\text{element} - \min_val]$ ga element qo'sh
6. $i = 0$
7. Har bir $\text{pigeonhole}[j]$ uchun:
 - $\text{pigeonhole}[j]$ dagi barcha elementlarni $\text{massiv}[i]$ ga yoz

- i ni oshir

Union-Find strukturasidan farqli o'laroq, Pigeonhole Sort qo'shimcha murakkab ma'lumot strukturalarini talab qilmaydi. Faqat oddiy massivlar va hisoblagichlar kifoya qiladi. Bu uni amalga oshirish va tushunish uchun juda sodda qiladi.

Algoritmning **matematik asosi** Kabutar uyasi printsipi va taqsimlash nazariyasiga asoslanadi. Har bir mumkin bo'lgan qiymat uchun joy ajratish orqali, saralash jarayoni solishtirishsiz amalga oshiriladi. Bu esa vaqt murakkabligini sezilarli darajada kamaytiradi va ba'zi hollarda optimal natijaga olib keladi.

Algoritmning afzalliklari va kamchiliklari

Pigeonhole Sort algoritmi o'zining noyob xususiyatlari tufayli bir qator muhim **afzalliklarga** ega. Eng muhim afzallik - **yuqori tezkorligidir**. Agar qiymatlar diapazoni (k) elementlar soniga (n) nisbatan kichik bo'lsa, algoritm $O(n)$ ga yaqin vaqtda ishlaydi. Bu xususiyat uni barcha solishtirishga asoslangan algoritmlardan ancha tez qiladi. Masalan, 1000 ta elementni 0-100 diapazonida saralash Quick Sort uchun $O(n \log n) \approx 10,000$ operatsiya talab qilsa, Pigeonhole Sort faqat 1,100 operatsiya bajaradi.

Ikkinchi muhim afzallik - **barqarorligi** (stability). Algoritm bir xil qiymatlarga ega elementlarning nisbiy tartibini saqlaydi, bu ko'plab amaliy vazifalar uchun muhim. Uchinchidan, **soddaligi** - algoritm tushunish va amalga oshirish uchun juda oddiy, kodni yozish va debug qilish oson. **Deterministik ishlash** ham muhim afzallik - bir xil kiritish uchun doimo bir xil natija beradi.

Shuningdek, algoritm **parallellashtirish** uchun yaxshi mos keladi. Taqsimlash bosqichini parallel ravishda amalga oshirish mumkin, bu esa multicore tizimlarda qo'shimcha tezlanish beradi. **Adaptivlik** ham foydali xususiyat - agar ma'lumotlar qisman saralangan bo'lsa ham, algoritm bir xil tezlikda ishlaydi.

Biroq algoritmning jiddiy **kamchiliklari** ham mavjud. Asosiy kamchilik - **xotira talabi**. Algoritm k o'lchamli qo'shimcha massiv talab qiladi, bu yerda k - qiymatlar diapazoni. Agar diapazon katta bo'lsa (masalan, 32-bit sonlar uchun 4



milliard), xotira muammosi yuzaga keladi. Ikkinchidan, **cheklangan qo'llanish sohasi** - algoritm faqat butun sonlar yoki cheklangan diapazonli ma'lumotlar bilan ishlaydi.

Siyrak ma'lumotlar uchun samarasiz - agar kichik miqdordagi element katta diapazonni egallasa, ko'p xotira behuda sarflanadi. Masalan, faqat 2 ta element (1 va 1,000,000) saralash uchun 1 million o'lchamli massiv kerak bo'ladi. **Float sonlar** bilan to'g'ridan-to'g'ri ishlay olmaydi, oldin butun sonlarga konvertatsiya qilish kerak.

Dinamik diapazon muammosi ham mavjud - agar yangi elementlar qo'shilsa va ular mavjud diapazondan tashqarida bo'lsa, butun struktura qayta tuzilishi kerak. **Cache locality** ham yomon bo'lishi mumkin, chunki xotiraning turli qismlariga tasodifiy kirishlar bo'ladi.

Solishtirishlar:

Quick Sort bilan: $O(n \log n)$ vs $O(n + k)$, lekin xotira $O(\log n)$ vs $O(k)$

Merge Sort bilan: Barqarorlik bir xil, lekin xotira talabi farqli

Counting Sort bilan: Juda o'xshash, lekin Pigeonhole Sort elementlar ro'yxatini saqlaydi

Radix Sort bilan: Radix Sort kattaroq diapazonlar uchun mos, Pigeonhole Sort kichikroq uchun

Umuman olganda, Pigeonhole Sort o'zining maxsus qo'llanish sohasida - kichik diapazonli, ko'p miqdordagi ma'lumotlar uchun - juda samarali, ammo umumiy maqsadli saralash algoritmi emas.

Amaliy qo'llanish sohalari

Pigeonhole Sort zamonaviy dasturlash va ma'lumotlarni qayta ishlashda juda ko'p amaliy qo'llanish topadi. Uning eng muhim qo'llanish sohalaridan biri **ta'lim tizimidagi baholarni saralash**dir. Maktab va universitetlarda talabalar baholarini (masalan, 0-100 diapazonida yoki A-F harfli baholar) saralashda Pigeonhole Sort optimal tanlov hisoblanadi. Minglab talaba bahosini bir necha millisekundda saralash mumkin.



O'yin dasturlashda algoritm keng qo'llaniladi. Onlayn o'yinlarda o'yinchilarning ball jadvallarini (leaderboard) saralash, darajalar bo'yicha tartiblash va turnir natijalarini ko'rsatish uchun ishlatiladi. **Statistik tahlillarda** yoshlar guruhlarini saralash, daromad darajalarini tartiblash va demografik ma'lumotlarni qayta ishlashda ham muhim rol o'ynaydi.

Veb-saytlar va mobil ilovalarda foydalanuvchilar reytinglarini, sharh balllarini va mahsulot baholarini saralashda Pigeonhole Sort tez-tez ishlatiladi. **E-commerce platformalarida** mahsulotlarni narx diapazoni, yulduzcha reytingi yoki kategoriya bo'yicha saralashda ham samarali.

Ma'lumotlar bazalarida indekslar yaratish va so'rovlarni optimallashtirish uchun qo'llaniladi. Ayniqsa, kategorik ma'lumotlar (holati, turi, prioriteti) bo'yicha saralashda foydali. **Real vaqt tizimlarida**, masalan, IoT sensorlar ma'lumotlarini qayta ishlash, tarmoq trafigini monitoring qilish va log fayllarni tahlil qilishda ham ishlatiladi.

Bioinformatikada DNK ketma-ketliklar, protein strukturalari va gen ekspressiya ma'lumotlarini saralashda Pigeonhole Sort qo'llaniladi. **Rasm va signal qayta ishlashda** piksel qiymatlarini (0-255 diapazonida) saralash, histogramm yaratish va rasm filtrlarini qo'llashda foydali.

Algoritmik savdo (algorithmic trading) da birja ma'lumotlarini real vaqtda saralash, narx darajalarini tartiblash va savdo signallarini qayta ishlashda ishlatiladi. **Kompyuter grafikasida** ranglar palitrasini saralash, Z-buffer algoritmlari va 3D rendering da ham qo'llaniladi.

Java kutubxonalarida Collections.sort() ichida maxsus hollarda Pigeonhole Sort ishlatiladi. **C++ STLda** std::sort() ham ba'zan kichik diapazonlar uchun shu usulni qo'llaydi.

Big Data muhitida Apache Spark va Hadoop kabi frameworklarda ma'lumotlarni taqsimlash bosqichida Pigeonhole Sort printsipi qo'llaniladi. **Machine Learningda** feature engineering jarayonida kategorik ma'lumotlarni qayta ishlashda ham foydali.



IoT va Edge Computingda cheklangan resurs va xotiraga ega qurilmalarda saralash vazifalarini tez va samarali bajarish uchun Pigeonhole Sort optimal tanlov hisoblanadi.

Xulosa

Pigeonhole Sort algoritmi zamonaviy dasturlashda o'zining maxsus o'rniga ega bo'lgan samarali saralash algoritmidir. Uning asosiy kuchi cheklangan diapazonli ma'lumotlar bilan ishlashda namoyon bo'ladi, bu yerda u $O(n + k)$ vaqt murakkabligi bilan boshqa barcha saralash algoritmlaridan ustun keladi. Algoritmning matemati asosi Kabutar uyasi printsipiga asoslanadi va taqsimlash usuli orqali solishtirishsiz saralashni amalga oshiradi.

Algoritmning asosiy **afzalliklari** - yuqori tezkorlik (optimal sharoitlarda), soddalik, barqarorlik va parallellashtirish imkoniyati. Biroq **cheklovlar** ham sezilarli: katta xotira talabi, cheklangan qo'llanish sohasi, siyrak ma'lumotlarda samarasizlik va faqat butun sonlar bilan ishlash imkoniyati. Bu xususiyatlar Pigeonhole Sort ni umumiy maqsadli emas, balki maxsus vazifalar uchun mo'ljallangan algoritm qiladi.

Amaliy qo'llanish sohalari juda keng: ta'lim (baholarni saralash), o'yin dasturlash (ball jadvallari), statistik tahlil, veb-dasturlash, ma'lumotlar bazalari, real vaqt tizimlari, bioinformatika va ko'plab boshqa sohalar. Algoritmning soddaligi uni o'rganish va amalga oshirish uchun ideal qiladi, shu bilan birga uning samaradorligi amaliy loyihalarda qimmatli qiladi.

Kelajak istiqbollari sifatida, Pigeonhole Sort parallel va taqsimlangan hisoblash muhitida yanada keng qo'llanilishi kutilmoqda. Big Data, IoT va edge computing texnologiyalarining rivojlanishi bilan algoritmgaga bo'lgan ehtiyoj ortib boradi. Quantum computing va neuromorphic architectures ham Pigeonhole Sort printsiplaridan foydalanish imkoniyatlarini ochmoqda.

Ta'lim jihatidan Pigeonhole Sort algoritm dizayni va tahlili o'rganish uchun mukammal misoldir. U talabalar va dasturchilarla vaqt-xotira almashish (time-space tradeoff) printsiplarini, maxsus algoritmlar yaratish usullarini va turli holatlarda eng mos algoritmni tanlash ko'nikmalarini o'rgatadi.





Tavsiyalar sifatida, dasturchilar va tadqiqotchilarga Pigeonhole Sort ni turli kontekstlarda sinab ko'rishlari, uni boshqa algoritmlar bilan solishtirishlari va o'zlarining maxsus vazifalariga moslashtirishlari tavsiya etiladi. Algoritmnini to'g'ri tanlov qilish - ya'ni qachon Pigeonhole Sort ishlatish kerak va qachon boshqa algoritmlar afzalroq ekanligi - muhim amaliy ko'nikmadir.

Pigeonhole Sort haqiqiy ma'noda "to'g'ri vaqtda, to'g'ri joyda" qo'llaniladigan algoritm. Uning o'rganilishi nafaqat konkret saralash muammosini hal qilish usulini beradi, balki algoritmik fikrlash, muammolarni tahlil qilish va samarali yechimlar topish ko'nikmalarini ham rivojlantiradi. Bu algoritm zamonaviy dasturlashning ajralmas qismi sifatida o'z ahamiyatini saqlab qoladi.

Foydalanilgan adabiyotlar

1. Knuth, Donald E. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Professional, 2nd Edition.
2. Thomas H. Cormen et al. *Introduction to Algorithms*. MIT Press, 4th Edition.
3. Robert Sedgewick, Kevin Wayne. *Algorithms*. Addison-Wesley, 4th Edition.
4. Weiss, Mark Allen. *Data Structures and Algorithm Analysis in Java*. Pearson, 3rd Edition.
5. Dasgupta, Sanjoy et al. *Algorithms*. McGraw-Hill Education.
6. Kleinberg, Jon; Tardos, Éva. *Algorithm Design*. Addison-Wesley.
7. Levitin, Anany. *Introduction to The Design and Analysis of Algorithms*. Pearson, 3rd Edition.
8. Goodrich, Michael T. et al. *Algorithm Design and Applications*. Wiley.
9. GeeksforGeeks. *Pigeonhole Sort Algorithm*.
<https://www.geeksforgeeks.org/>
10. Wikipedia. *Pigeonhole Sort*. https://en.wikipedia.org/wiki/Pigeonhole_sort
11. LeetCode Discussions. *Sorting Algorithms Comparison and Analysis*.
12. Stack Overflow. *When to use Pigeonhole Sort vs other sorting algorithms*.
13. Algorithm Visualizer. *Pigeonhole Sort Step-by-Step Visualization*.
14. MIT OpenCourseWare. *Introduction to Algorithms - Sorting*.
15. Princeton University. *Algorithms and Data Structures Course Materials*.