Статистическо изследване върху алгоритми за сортиране

Камен Димитров Младенов

23 Януари 2021

https://github.com/Syndamia/latex-projects/tree/main/Statistical%20analysis%20of%20sorting%20algorithms

Съдържание

1. Същност и избор на темата

Сортиращият алгоритъм е един от най-важните аспекти на програмирането в днешно време. Чрез него, дадени елементи могат да бъдат подредени в нарастваща (или намаляваща) последователност, което намира употреба във всичко от новинарски сайтове и социални медии, до видео игри и операционни системи.

Този тип алгоритъм бива изследван практически от началото на компютърната наука - най-ранните сортиращи алгоритми се забелязват през 1951, а анализиране на Bubble Sort, може би най-известния сортиращ алгоритъм, бива извършено още през 1957.[3]

В днешно време има десетки и стотици алгоритми, което дълбоко усложнява изборът на обикновения програмист. Затова тази статия се стреми да сравни главните им характеристики.

2. Избор на алгоритми и разглеждани характеристики

Избраните алгоритми за сравнение са подбрани от Toptal[2]. Този сайт не е в никакъв случай решаващо място за най-известни алгоритми, но за целите на този документ е достатъчен.

Избраните алгоритми са:

- Shell Sort
- Merge Sort
- Heap Sort
- Quick Sort

Данни относно характеристиките им ще бъдат взети от Toptal[2] и от warp.povusers.org[1]. Отново, тези сайтове не са решаващи източници на която и да е от данните, но са достатъчни.

Избрани характеристики:

- Сложност
- Скорост
- Брой сравнения
- Брой записвания

3. Представителна извадка

Разглеждайки дадените източници, можем да извадим следните стойности:

Име на алгоритъм	Сложност	Скорост	Брой сравнения	Брой записвания
Shell Sort	6666	0.71	101319	136506
Merge Sort	18494	0.67	56823	70000
Heap Sort	18494	0.59	107688	171318
Quick Sort	18494	0.60	95321	42888

Важни бележки относно вземането на тези данни:

- сложността е изчислена при горната граница (O) с 5000 елемента
- скоростта, броят сравнения и броят записвания са взети при сравнение на 5000 случайно разбъркани целочислени числа с минимални повторения помежду им[1]
- скоростта (както записано в warp.povusers.org[1]) е в милисекунди
- в ситуации на неизвестна или променлива абсолютна максимална сложност, се използва тази сложност, която дава най-висока стойност

4. Статистика на данните

Снабдени с данни, вече можем да правим статистически анализ върху тях. Следващите подраздели показват с таблици и диаграми различни свойства и съотношения на тази информация.

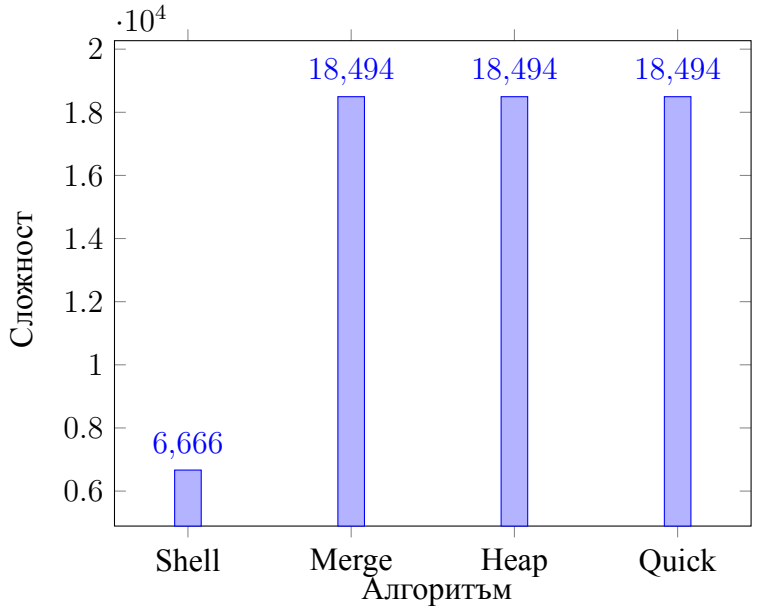
4.1. Вариационни редове

Сложност	Shell sort 6666	Merge Sort 18494	Heap Sort 18494	Quick Sort 18494
Скорост	Heap sort 0.59	Quick Sort 0.60	Merge Sort 0.67	Shell Sort 0.71
Брой сравнения	Merge sort 56823	Quick Sort 95321	Shell Sort 101319	Heap Sort 107688
Брой записвания	Quick sort 42888	Merge Sort 70000	Shell Sort 136506	Heap Sort 171318

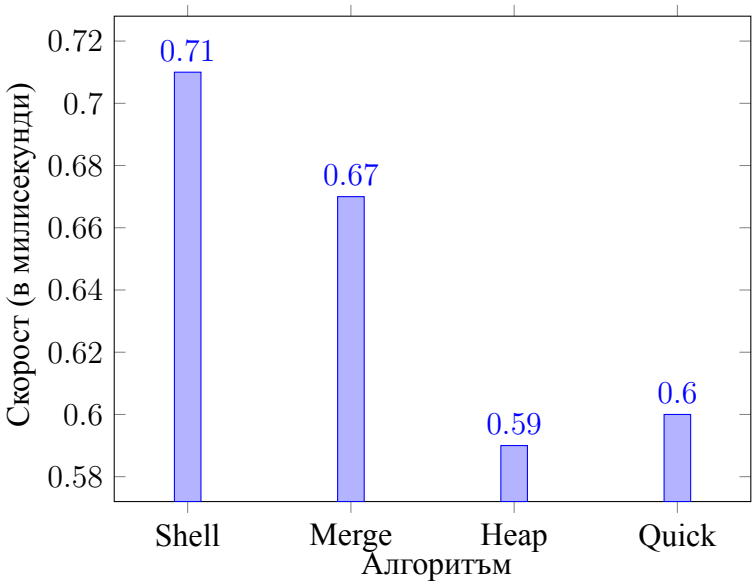
4.2. Мода, медиана, средна стойност

Характеристика	Мода	Медиана	Средна стойност
Сложност	18494	$(18494 + 18949)/2 = \underline{18494}$	15537
Скорост	Няма	$(0.60 + 0.67)/2 = \underline{0.635}$	0.6425
Брой сравнения	Няма	$(95321 + 101319)/2 = \underline{98320}$	90287.75
Брой записвания	Няма	$(70000 + 136506)/2 = \underline{103253}$	105178

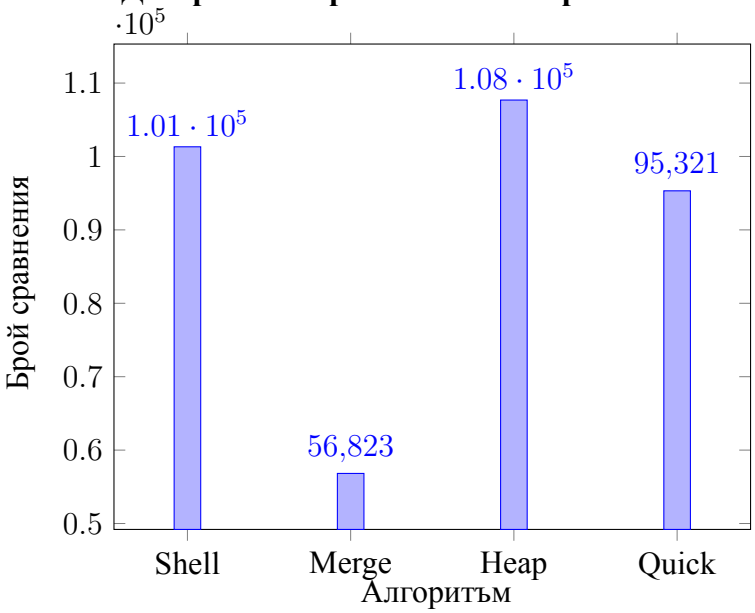
4.3. Сравнение на данни



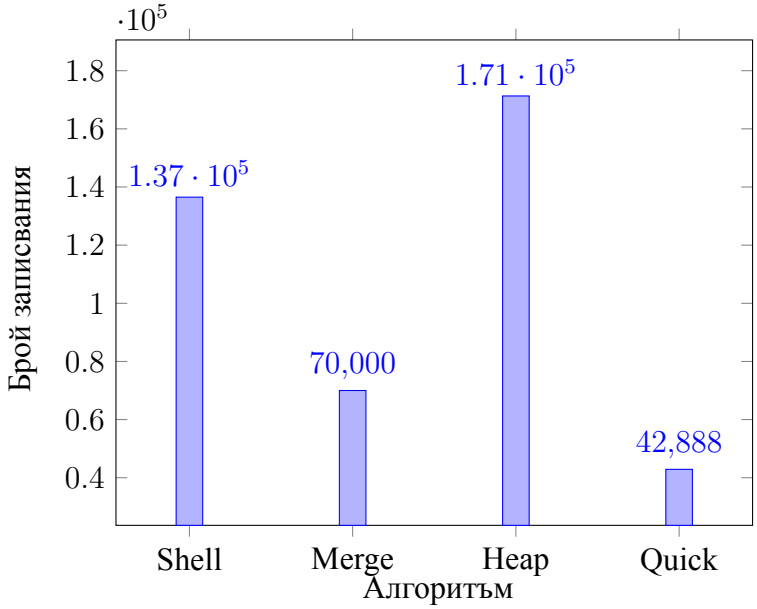
Диаграма 1: Сравнение на сложност



Диаграма 2: Сравнение на скорост



Диаграма 3: Сравнение на брой сравнения



Диаграма 4: Сравнение на брой записвания

5. Заключение и анализ на резултатите

Получената информация дава интересен, но дълбоко непълен, поглед към ефикасността на тези четири алгоритъма. Разбира се, данните са събрани при само един тест в само една ситуация, затова каквото заключение да направим, то не бива да бъде взето за дадено.

Quick Sort не е най-бързият алгоритъм, макар и думата "бърз" да фигурира в името му. Тази титла взема Heap Sort, който има същата сложност, обаче му трябват повече сравнения и записвания.

Интересно наблюдение е как сложността на Shell Sort е почти три пъти по-малка от другите три алгоритъма, ала изисква най-много сравнения и записвания, и е вторият по бавност.

Обаче, в края на деня, се забелязва най-важното нещо: всеки алгоритъм има своя роля. Най-прост е Shell, най-бърз е Heap, най-малко сравнения ползва Merge и най-малко записвания прави Quick. Правилния начин на избор на един от тези алгоритми е анализирането на ситуацията, в която той ще бъде използван.

Но, все пак, бих желал да отбележа, че не можеш да сгрешиш, ако ползваш Quick Sort. Той се класира на второ или първо място във всички разглеждани категории, а в много от тези, които не сме разглеждали, той побеждава всяка конкуренция!

Източници

[1] Comparison of several sorting algorithms: Integers - http://warp.povusers.org/sortcomparison/integers.html.

[2] Sorting algorithms - https://www.toptal.com/developers/sorting-algorithms.

[3] Sorting algorithm history - https://en.wikipedia.org/wiki/sorting_algorithm#history, Dec 2020.