

режимі чернетки. Для цього потрібно активувати режим читання або друку, який відображає окремі сторінки з видимими колонтитулами, але, на жаль, редагувати документ у цьому режимі перегляду недоступно.

За замовчуванням у Google Docs навпаки. Тут присутні окремі текстові сторінки з верхнім і нижнім колонтитулами прямо перед собою, ніби сторінки були надруковані. У налаштуваннях є “Змінити вид на чернетку”. На відмінну від Word Online, тут можна редагувати в обох режимах.[3]

Таким чином Google Docs має перевагу в цьому аспекті.

У підсумку, обидва інструменти функціонально дуже схожі, основними винятками є панель інструментів швидкого редагування, яка доступна лише у Word Online, і більш широкі параметри форматування зображень і функції коментарів. Але, якщо придивитися ближче, Google Docs трохи випереджають.

Деякі відсутні або обмежені функції, зокрема історія версій і режим виправлення, доступні в локально встановленій програмі Word, але недоступні в онлайн-версії. Це створює відчуття непослідовності для користувачів і робить Word Online менш ефективним для онлайн-співпраці. Крім того, проблеми сумісності між різними версіями автономної версії Word можуть спричинити проблеми, тоді як у чистих онлайн-інструментах, таких як Google Docs, усі користувачі автоматично працюють з тією самою (і завжди оновленою) версією.

Список літератури

1. Служба підтримки Microsoft URL: <https://support.microsoft.com/uk-ua>
2. Google Docs Editors Helps URL: https://support.google.com/docs/answer/9228272?hl=en&ref_topic=9045928
3. Google Docs vs. Microsoft Word: Which One Is Suitable for You? URL: <https://softwarekeep.com/blog/google-docs-vs-microsoft-word>

УДК 004.021

*Підруцький Д.А., студент 2 курсу
Спеціальності 122«Комп'ютерні науки»
Ніколюк П.К., професор, доктор
фізико-математичних наук
кафедри Інформаційних технологій*

ОСНОВНІ АЛГОРИТМИ СОРТУВАННЯ

Донецький національний університет імені Василя Стуса

В сучасному світі використання алгоритмів є обов'язковим, якщо ви хочете здобути навички чи навчитись чомусь, оскільки це слідування якомусь плану. Та офіційно це набір інструкцій, які описують порядок дій виконавця, для того щоб досягти результату розв'язання задачі за скінченну кількість дій чи система правил виконання конкретного процесу, яка досягає поставленої мети за

скінченну кількість часу[1]. Також доволі часто алгоритми описують в вигляді блок-схем, оскільки дана форма є доволі зручною для розуміння.

Тож в даній роботі, я хочу розповісти про алгоритми аналізу та обробки даних, в вигляді сортування даних, саме в тематиці програмування, які як і будь які інші алгоритми представляють собою порядок дій і методів, за допомогою яких відбувається впорядкування списку предметів, наприклад масивів з програмування.

Хоч і задача виглядає простою, оскільки потрібно просто впорядкувати елементи, але на разі є доволі багато алгоритмів складних і швидких чи простих і повільних. Хоч і основними розбіжностями між ними є лиш час виконання $O(n^2)$, необхідність в додатковій пам'яті, яка найчастіше є такою – $O(1)$ і стабільність, при якій не змінюється розташування елементів з однаковими ключами.

Однаке зазвичай основним показником таких алгоритмів є саме час виконання $O(n^2)$, згідно якого ми напряду можемо побачити наскільки швидким, використаний алгоритм може бути і який ми будемо використовувати для розрізнення алгоритмів.

Почнемо з найпростішого $O(1)$ кількість операцій, якого, а отже і важкість збільшується разом зі збільшенням кількості введених елементів при сортуванні[2].

Далі по складності йдуть лінійні алгоритми $O(n)$. Прикладом такого алгоритму, є сортування підрахунком, який як не дивно, полягає в підрахунку разів, коли кожен елемент зустрічається в вихідному масиві, а потім за 1 прохід поставити всі елементи на свої місця. В загальному, такий алгоритм набагато ефективніше $O(n)$, до того ж він виконується в «постійному/постійному часі», тобто це $O(1)$. Насправді операція не одна: потрібно отримати довжину масиву, отримати останній елемент, виконати множення і ділення. Що схоже на $O(1)$ алгоритми, але фактична кількість кроків не має значення, а саме важливо в ньому, щоб алгоритм виконувався в постійному часі. Прикладом якого є сортування підрахунком.

До алгоритмів зі складністю $O(n^2)$, зазвичай відносять алгоритми із вкладеними циклами для кожного елемента списку. Прикладом найпростішого методу цієї складності є сортування бульбашкою, який через цю ж простоту, вважається навчальним і практично не використовується поза навчальною літературою, замість цього на практиці використовуються більш ефективні алгоритми сортування[3].

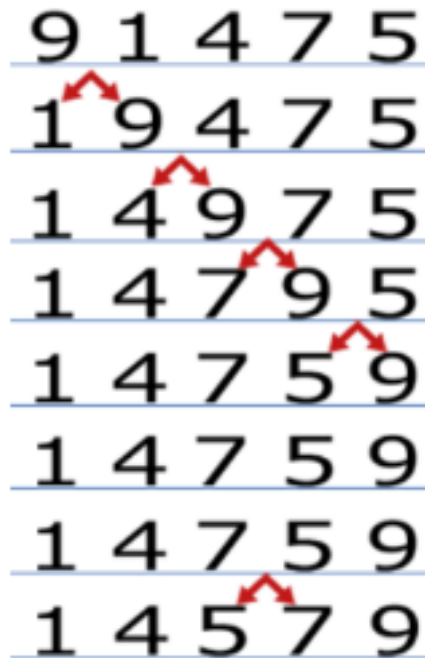


Рис.1 Робота алгоритму бульбашки

Роботу якого по пересуванні кожного елемента, ми чітко бачимо, на кожному кроці ілюстрації Рис.1. Робота алгоритму бульбашки

Далі на черзі в нас логарифмічне сортування $O(n \log n)$, де серед найпоширеніших алгоритмів цього типу відносять пірамідальне сортування, яке гарантовано працює за $O(n \log n)$ операцій при сортуванні n елементів, а кількість пам'яті не залежить від розміру масиву тобто, $O(1)$ а також він є не стабільним[4].

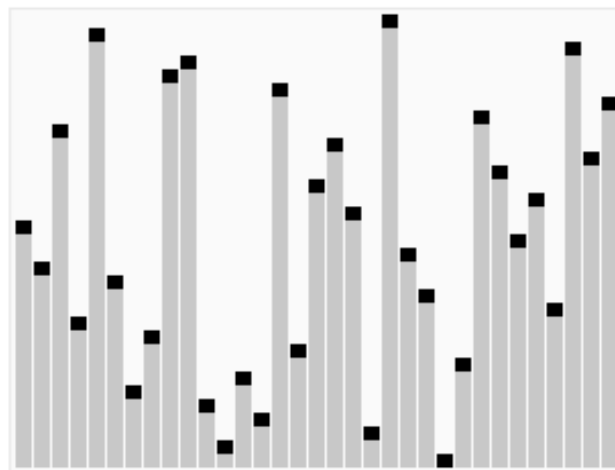


Рис.2 Робота пірамідального алгоритму

Роботу якого, по сортуванню елементів, ми бачимо на Рис.2 Робота пірамідального алгоритму

Далі продивимось алгоритм $O(n \log^2 n)$, який на справді є трохи ускладненою версією минулого алгоритму $O(n \log n)$, до прикладу якого візьмемо алгоритм Шелла, який представляє собою узагальнене сортування включенням $O(n^2)$, але, яке переміщує елемент тільки на одну позицію за раз.

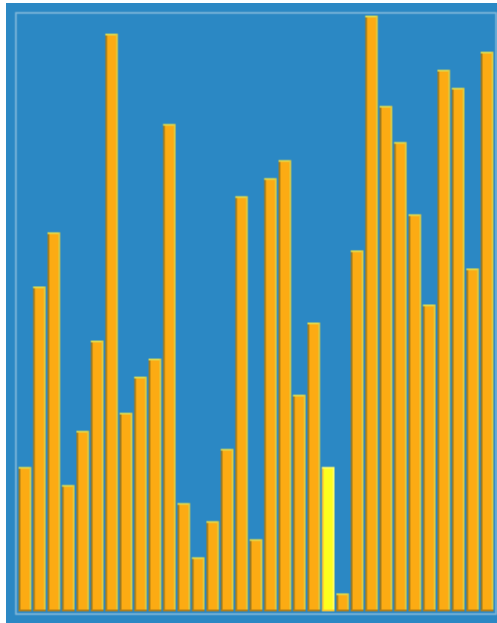


Рис.3 Робота алгоритму Шелла

Роботу якого, ми наглядно спостерігаємо на Рис.3 Робота алгоритму Шелла

І нарешті, найскладнішим алгоритмом є $O(n!)$, прикладом якого є випадкове сортування, яке насправді є менш ефективним, а також згідно Вікіпедії [5], має не найкращу славу, але найкращий час виконання $O(n)$ і наочно використовується для впорядкування колоди карт таким чином, по такому механізму, що спочатку уявна колода підкидається у повітря, карти збираються у довільному порядку і перевіряється, чи є колода впорядкованою. А якщо колода невпорядкована, то операцію повторюють, до поки наша колода, або список не буде впорядкованим.

В підсумку, всі алгоритми мають свою цінність і призначення в загальному плані від найпростішого до найскладнішого, але за допомогою всіх, хоч і різними методами, ми можемо досягти заданої мети.

Список літературних джерел:

[1] WIKIPEDIA URL:

https://uk.wikipedia.org/wiki/Алгоритм_сортування (дата звернення: 28.11.2022)

[2] Хабр URL:

<https://habr.com/ru/post/444594/> (дата звернення: 28.11.2022)

[3] WIKIWAND URL:

https://www.wikiwand.com/uk/Сортування_бульбашкою (дата звернення: 28.11.2022)

[4] WIKIPEDIA URL:

https://uk.wikipedia.org/wiki/Пірамідальне_сортування (дата звернення: 28.11.2022)

[5] WIKIPEDIA URL:

https://uk.wikipedia.org/wiki/Випадкове_сортування (дата звернення: 28.11.2022)