

Карташов Пилип,

студент 2 курсу

Факультету фізико-математичної, комп'ютерної та технологічної освіти

Наук. керівник: **Н.В. Кравченко,**

к.ф.-м.н., доцент (БДПУ)

ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА СКЛАДНОСТІ АЛГОРИТМУ

Актуальність теми. Складність алгоритмів зазвичай оцінюють за часом виконання або за використаною пам'яттю. В обох випадках складність залежить від розмірів вхідних даних: масив з 100 елементів буде оброблений швидше, ніж аналогічний з 1000. При цьому точний час мало кого цікавить: воно залежить від процесора, типу даних, мови програмування і інших параметрів. Важлива лише асимптотична складність, тобто складність при прагненні розміру вхідних даних до нескінченності.

Ступінь досліджуваності проблеми. Проблематиці оцінки складності алгоритму приділили увагу вітчизняні науковці В.В. Корнеєв, А.Ф. Гареев, С.В. Васютин та багато інших [1].

Мета та методи дослідження. Визначити експериментально характеристики алгоритмів різної складності. Методи – аналіз, узагальнення, експеримент.

Сутність дослідження. Припустимо, певним алгоритмом потрібно виконати $4n^3 + 7n$ умовних операцій, щоб обробити n елементів вхідних даних. При збільшенні n на підсумковий час роботи буде значно більше впливати зведення n в куб, ніж множення його на 4 або ж доповнення $7n$. Тоді кажуть, що тимчасова складність цього алгоритму дорівнює $O(n^3)$. Оцінка складності алгоритму позначається літерою O . Формально $O(f(n))$ означає, що час роботи алгоритму (або обсяг займаної пам'яті) росте в залежності від обсягу вхідних даних не швидше, ніж деяка константа, помножена на $f(n)$.

Види складності алгоритмів:

Складність $O(1)$. Час роботи алгоритму взагалі не залежить від розміру вхідних даних. Наприклад, для визначення значення третього елемента масиву не потрібно ні запам'ятовувати елементи, ні проходити по ним скільки-то раз. Завжди потрібно просто дочекатися в потоці вхідних даних третій елемент і це буде результатом, на обчислення якого для будь-якої кількості даних потрібно один і той же час.

$O(n)$ – лінійна складність. Такою складністю володіє, наприклад, алгоритм пошуку найбільшого елемента в невідсортованому масиві. Нам доведеться пройти по всіх n елементів масиву, щоб зрозуміти, який з них максимальний.

$O(\log n)$ – логарифмічна складність. Найпростіший приклад – бінарний пошук. Якщо масив відсортований, ми можемо перевірити, чи є в ньому якесь конкретне значення, шляхом розподілу навпіл. Перевіримо середній елемент, якщо він більше пошукового, то відкинемо другу половину масиву – там його точно немає. Якщо ж менше, то навпаки –

відкинемо початкову половину. І так будемо продовжувати ділити навпіл, в результаті перевіримо $\log n$ елементів.

$O(n^2)$ – квадратична складність. Таку складність має, наприклад, алгоритм сортування вставками. Він вдає із себе два вкладених цикли: один, щоб проходити по всьому масиву, а другий, щоб знаходити місце чергового елементу для вставки в уже відсортованій частині. Таким чином, кількість операцій буде залежати від розміру масиву як $n * n$, тобто n^2 .

Для експериментальної перевірки наведених алгоритмів було реалізовано програми з різними видами складності та визначено час виконання цієї програми на одному комп'ютері.

1. Задача 1 з лінійною складністю – виведення найбільшого елемента наперед заданого масиву з 30 елементів.

2. Задача 2 зі складністю $O(n)$ – виведення п'ятого елемента наперед заданого масиву з 30 елементів.

3. Задача 3 з логарифмічною складністю $O(\log n)$ – виведення номера комірки, в якій знаходиться число 13 наперед заданого масиву з 30 елементів (від 1 до 30) за допомогою бінарного пошуку.

4. Задача з квадратичною складністю $O(n^2)$ – сортування вставками за зростанням наперед заданого масиву з 30 елементів.

Оцінка складності алгоритмів:

Основні висновки. Всі задачі мають однакову пам'ять, при цьому задача 2 зі складністю $O(n)$ виконується за 0.1041с. задача 1 лінійної складності – на 0.04с. повільніше (0.1433 с.), задача 3 з логарифмічною складністю – на 0.1с. повільніше (0.1579с.), задача 4 з квадратичною складністю – ще на 0.1с. повільніше (0.1693с.).

ЛІТЕРАТУРА

1. Нікітченко М. С., Шкільняк С. С. Математична логіка та теорія алгоритмів //К.: ВПЦ Київський університет. – 2008.

Клименко Антон,

студент 2 курсу

Факультету фізико-математичної, комп'ютерної та технологічної освіти

Наук. керівник: **Н.В. Кравченко,**

к.ф.-м.н., доцент (БДПУ)

АУДІОЛЕКЦІЇ В ІНКЛЮЗИВНІЙ ПРОФЕСІЙНІЙ ОСВІТІ

На сьогодні в Україні понад 60 тисяч осіб є інвалідами по зору, із них близько 25 тисяч осіб, що повністю втратили зір. У всесвітньому масштабі ці цифри просто вражають. Ці люди стикаються з труднощами в реалізації своїх прав, вони позбавлені багатьох можливостей реалізувати права на одержання інформації. Оскільки вони не мають можливості належно забезпечити себе необхідними інформаційними ресурсами, наприклад, книгами в шрифтах Брайля чи аудіо-літературою.

Теперішній стан забезпечення незрячих людей літературою в Україні не може не хвилювати, адже в порівнянні з Європою чи США, у нас практично не видається література для незрячих. Якщо говорити про цифри, то це приблизно 5-6 книг шрифтом Брайля та біля 150 аудіо-книг на