

Алгоритми та структури даних

Підсумок першого семестру

Задачі

- пошук у масиві даних
- структурований доступ до даних
- сортування масиву даних
- пошук елемента за ключем
- пошук мінімуму/максимуму серед масиву даних

Пошук у масиві даних

- Перевірити, чи є елемент у масиві даних
- Знайти індекс елемента у масиві даних

	Лінійний пошук (Linear search)	Бінарний пошук (Binary search)
Найкраща складність	$O(1)$	$O(1)$
Середня складність	$O(N)$	$O(\log(N))$
Найгірша складність	$O(N)$	$O(\log(N))$

Структурований доступ до даних

Динамічний масив - структура, масив, що автоматично збільшує або зменшує розмір виділеної під нього пам'яті в залежності від кількості елементів

Переваги	Недоліки
Складність доступу до i -го елементу та запис у i -ий елемент за $O(1)$	Складність вставки або видалення елементів за $O(N)$ (враховуючи зсуви)
Складність вставки нових елементів за $O(1)$ (амортизована)	Доволі багато зайвого місця, яке не використовується (кількість виділеної пам'яті має бути більшою за поточний розмір масиву)

Структурований доступ до даних

Зв'язний список - структура, яка має загальний вигляд **[head, [tail]]**, де **head** — це дані, а **tail** — інший зв'язний список

Переваги	Недоліки
Не потребує зміни розміру, як динамічний масив, і не використовує зайвої пам'яті	Повільна (за $O(N)$) складність доступу до i -го елементу
	Зв'язний список розкиданий по пам'яті

Структурований доступ до даних

Черга — структура, яка реалізовує принцип FIFO – First In, First Out (елемент, який першим зайшов у структуру, перший з неї вийде). Приклад — жива черга

Стек — структура, яка реалізовує принцип LIFO – Last In, First Out (елемент, який останнім зайшов у структуру, перший з неї вийде). Приклад — стопка тарілок

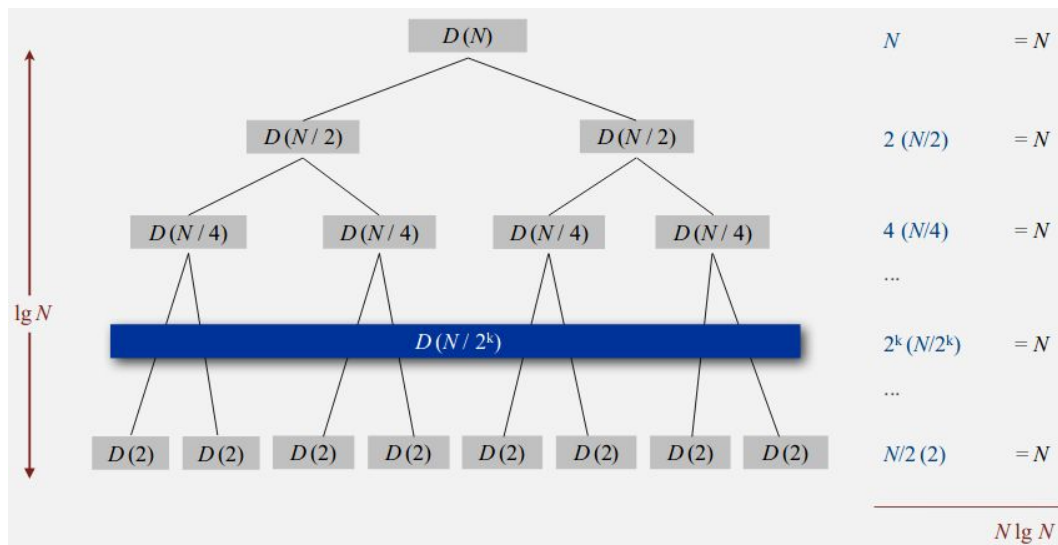
Сортування масиву даних

Елементарні сортування

Сортування	Найкращий випадок	Середній випадок	Найгірший випадок
Бульбашкою (Bubble)	$O(N)$	$O(N^2)$	$O(N^2)$
Вибором (Selection)	$O(N^2)$	$O(N^2)$	$O(N^2)$
Вставкою (Insertion)	$O(N)$	$O(N^2)$	$O(N^2)$

Сортування масиву даних

Сортування злиттям (Merge Sort) — найкращий, середній та найгірший випадки — $O(N \cdot \log(N))$. Два підвиди — зверху вниз (top-down) та знизу вгору (bottom-up)



Сортування масиву даних

Швидке сортування (Quick sort)

- Найкращий випадок — $O(N \cdot \log(N))$
- Середній випадок — $O(N \cdot \log(N))$
- Найгірший випадок — $O(N^2)$

На відміну від сортування злиттям, не потребує додаткової пам'яті, а також на практиці швидший, адже виконує менше перестановок даних.

Однак, швидке сортування — нестабільне

Сортування масиву даних

Сортування	Найкращий випадок	Середній випадок	Найгірший випадок	Пам'ять	Стабільне
Бульбашкою (Bubble)	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N) + O(1)$	Так
Вибором (Selection)	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N) + O(1)$	Ні
Вставкою (Insertion)	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N) + O(1)$	Так
Злиттям (Merge)	$O(N \cdot \log(N))$	$O(N \cdot \log(N))$	$O(N \cdot \log(N))$	$O(N) + O(N)$	Так
Швидке (Quick)	$O(N \cdot \log(N))$	$O(N \cdot \log(N))$	$O(N^2)$	$O(N) + O(1)$	Ні
Підрахунком (Count)	$O(N + K)$	$O(N + K)$	$O(N + K)$	$O(N + K)$	Так
Радіальне (Radix, LSD)	$O(W \cdot N)$	$O(W \cdot N)$	$O(W \cdot N)$	$O(W + N)$	Так

Пошук елемента за ключем

Символьна таблиця

Абстракція пари ключ-значення — **вставити** в структуру **ключ** з певним **значенням**.

Гешування — перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Такі перетворення також називаються геш-функціями, або функціями згортання, а їхні результати називають гешем, геш-кодом, геш-сумою, або дайджестом повідомлення (англ. message digest)

Пошук елемента за ключем

Реалізації символьної таблиці	Середній випадок		Найгірший випадок	
	Пошук	Вставка	Пошук	Вставка
Зв'язний список / динамічний масив	$N / 2$	N	N	N
Впорядкований масив	$\log N$	$N / 2$	$\log N$	N
Геш-таблиця (метод ланцюжків)	constant (*)	constant (*)	N	N
Геш-таблиця (відкрита адресація)	constant (*)	constant (*)	N	N

Пошук мінімуму/максимуму серед масиву даних

- Лінійний пошук / бінарний пошук
 - Лінійний: побудова за $O(N)$, пошук \min/\max за $O(N)$, вставка нових елементів $O(N)$
 - Бінарний: побудова за $O(N \cdot \log(N))$, пошук \min/\max за $O(\log(N))$, вставка нових елементів за $O(N)$
- Купа (Heap)
 - Побудова: $O(N)$ (амортизована складність)
 - Пошук: $O(\log(N))$
 - Вставка нових елементів за $O(\log(N))$
 - Існують різновиди купи, такі як \max та \min купи