

Списки

Підготував: Скоренок Олександр

Список - абстрактний тип даних, який являє собою зліченне число впорядкованих значень, де одне і теж саме значення може зустрічатися більше одного разу

По суті він визначає послідовний набір елементів, до яких ви можете додавати нові елементи та видаляти або змінювати існуючі.

Структура даних, як правило, має дві дуже відмінні реалізації - **список-масив** (*англ. array list*) та **зв'язаний список** (*англ. linked list*).

Списки також є основою для інших абстрактних типів даних, зокрема черг, стеків і їх варіацій.

Список-масив - це, в по суті, масив, що самостійно змінює розмір, або, іншими словами, **динамічний масив**.

Ця структура даних поводить себе точно так само, як звичайний масив, але з додатковою властивістю ємності, яка полягає у збільшенні розміру кожного разу, коли його перевищують.

24	69	7	0
----	----	---	---

24	69	7	22
----	----	---	----

24	69	7	22	6	0	0	0
----	----	---	----	---	---	---	---

Складність

- Додати (елемент) - $O(1)$
- Видалити (елемент) - $O(n)$
- Вставити (елемент, індекс) - $O(n)$
- Отримати (індекс) - $O(1)$

Використання

Потреба в цій структурі даних найбільш очевидна, коли вам хочеться зберегти дані, не знаючи, скільки місця вам потрібно заздалегідь.

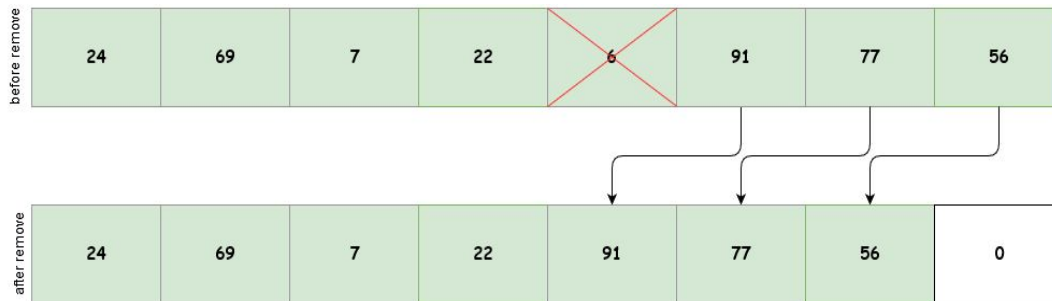
Також, коли потрібно швидко звертатись до елементів, або додавати.

Якщо потрібно часто видаляти або вставляти елементи, це невдалий вибір.

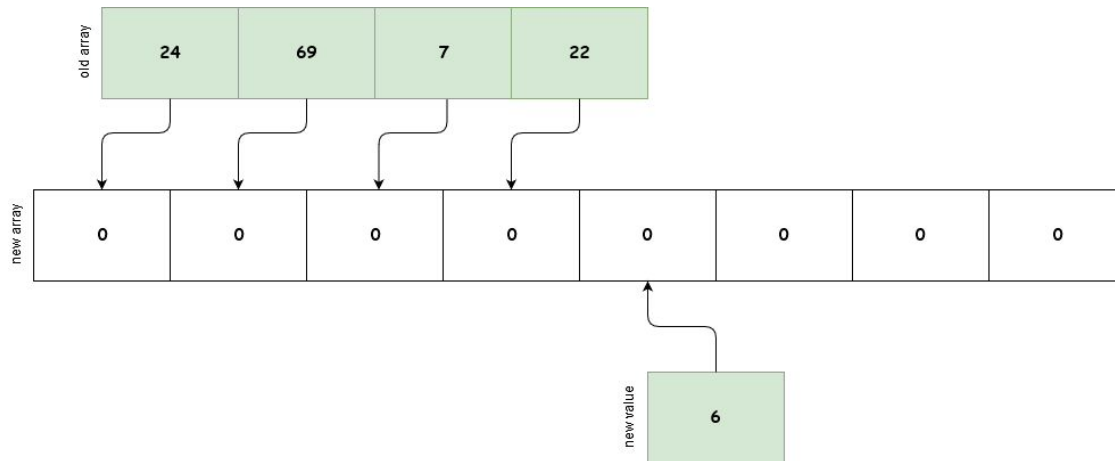
Реалізація

```
template<class T>
class ArrayList {
public:
    ArrayList();
    explicit ArrayList(int capacity);
    void PushBack(const T& item);
    void Insert(const T& item, int index);
    void RemoveAt(int index);
    T Get(int index) const;
private:
    void Resize();
    T* _array;
    int _capacity;
    int _size;
};
```

Операція **RemoveAt**



Операція **Resize**

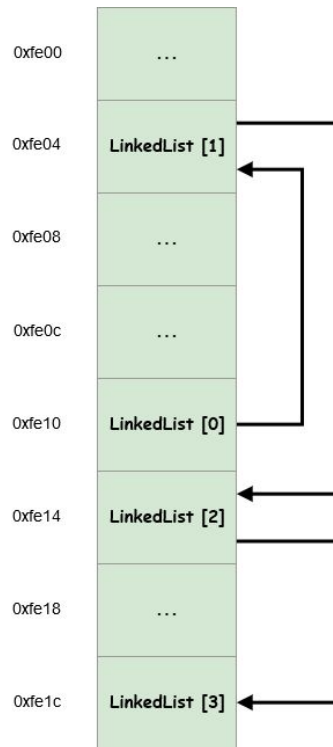


Зв'язаний список - це структура даних, ідея якої дуже схожа на список-масив, але з тією різницею, що її елементи зберігаються абсолютно по-іншому.

Елементи цієї структури даних розподілені по всій пам'яті, на відміну від списку масивів, в якому вони розташовані послідовно.



VS.



Складність

- Додати (елемент) - $O(1)$
- Видалити (елемент) - $O(n)$
- Вставити (елемент, індекс) - $O(n)$
- Отримати (індекс) - $O(n)$

Використання

Зв'язані списки добре підходять для програм, які потребують часто додавати або видаляти елементи з початку або з кінця набору елементів.

Та це невдалий вибір, якщо потрібно часто видаляти, отримувати або вставляти елементи.

Реалізація

```
template <typename T>
class LinkedList {
public:
    struct Node {
        T value;
        Node* next = nullptr;
    };
    ~LinkedList();
    void PushFront(const T& value);
    void InsertAfter(Node* node, const T& value);
    void RemoveAfter(Node* node);
    void PopFront();
private:
    Node* head = nullptr;
};
```

Порівняння списку-масиву та зв'язного списку

- Список-масив швидко додає та отримує елементи, тоді як зв'язаний список може швидко додавати або видаляти елементи на початку та в кінці структури даних.
- Зв'язаний список немає проблем із фрагментацією оперативної пам'яті, на відміну від його альтернативи - списку-масиву.
- Зв'язаний список використовує стільки пам'яті, скільки йому потрібно, на відміну від списку-масиву, який виділяє додаткову пам'ять.