

Структура даних FIFO



"First in? First out!"



- Н.В. Сокур

Лінійний список – послідовність $n \geq 0$ вузлів $X[1], X[2], \dots, X[n]$, головною структурною особливістю якої є умова, що коли $n > 0$ та $X[1]$ – перший вузол, а $X[n]$ – останній, то k -ий вузол $X[k]$ знаходиться за $X[k - 1]$ та до $X[k + 1]$ для всіх $1 < k < n$.

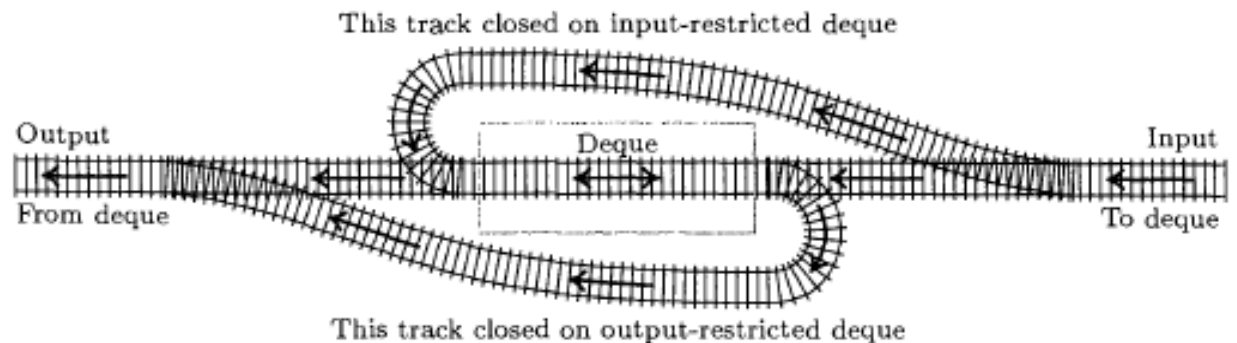
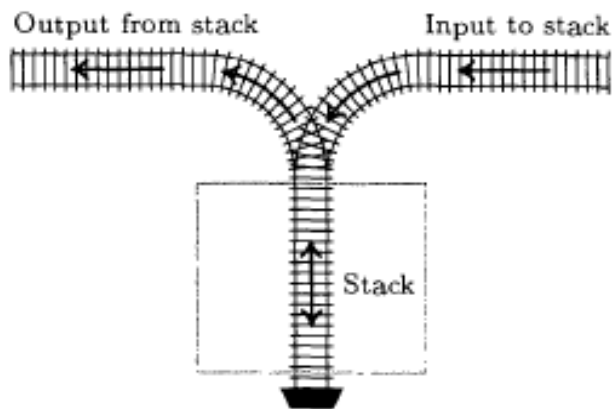
З лінійними списками можна виконувати наступні операції:

- i. Отримання доступу до k -ого вузла списка.
- ii. Вставка нового вузла відразу за або до k -го вузла.
- iii. Видалення k -го вузла.
- iv. Об'єднання в єдиний список двох (або більше) лінійних списків.
- v. Розбиття лінійного списка на два (або більше) списки.
- vi. Створення копії лінійного списка.
- vii. Визначення кількості вузлів списка.
- viii. Сортування вузлів в порядку збільшення значень в певних полях цих вузлів.
- ix. Пошук вузла із заданим значенням в певному полі.

Стек – лінійний список, в якому всі операції вставки та видалення (зазвичай і операції доступу до даних) виконуються лише на одному з кінців списку.

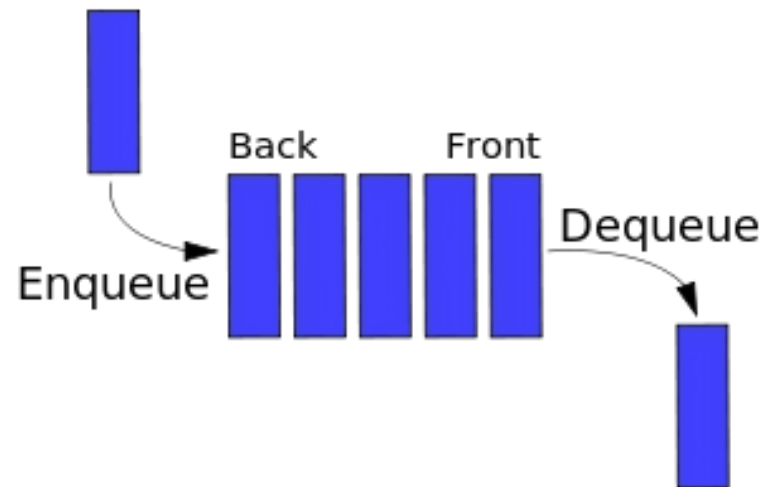
Черга або одностороння черга – лінійний список, в якому всі операції вставки виконуються на одному з кінців списку, а всі операції видалення (зазвичай і операції доступу до даних) – на іншому.

Дек або двостороння черга (double-ended queue) – лінійний список, в якому всі операції вставки та видалення (зазвичай і операції доступу до даних) виконуються на обох кінцях списку.

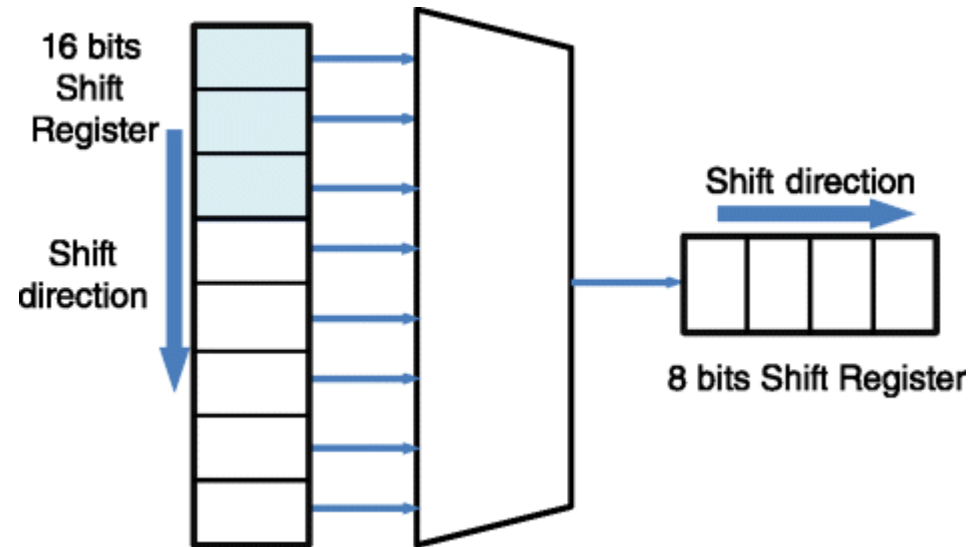


Черги часто називають циклічними сховищами (circular stores) або дисциплінами обслуговування в порядку надходження (first-in-first-out lists – FIFO lists).

Є кілька ефективних реалізацій черг FIFO. Ефективною реалізацією є така, що здатна виконувати операції – додати в чергу (enqueueing) та видалити з черги (dequeueing) – протягом часу $O(1)$.

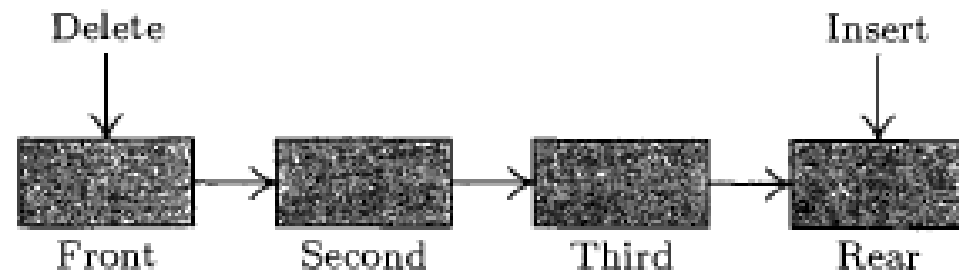


Концепцію FIFO можна реалізувати у вигляді апаратного реєстру зсуву (hardware shift register)



Або з використанням структур пам'яті:

- Зв'язний список
 - Двозв'язний список
 - Простий (однозв'язний) список
- Двостороння черга (дек)



(b) Queue

Реалізація зв'язаного списку FIFO

```
#include <iostream>
#include <stdexcept>

template <typename T>
class FIFO
{
private:

    struct Node {
        T      value;
        Node *next;

        Node(T _value) : value(_value), next(NULL) {}
    };

    Node *front;
    Node *back;
```

```
public:
    FIFO() : front(NULL), back(NULL) {}

    ~FIFO() {
        while (front != NULL)
            dequeue();
    }

    void enqueue(T _value) {
        Node *newNode = new Node(_value);

        if (front == NULL)
            front = newNode;
        else
            back->next = newNode;

        back = newNode;
    }

    T dequeue() {
        if (front == NULL)
            throw std::underflow_error("Nothing to dequeue");

        Node *temp = front;
        T      result = front->value;

        front = front->next;
        delete temp;

        return result;
    }
};
```