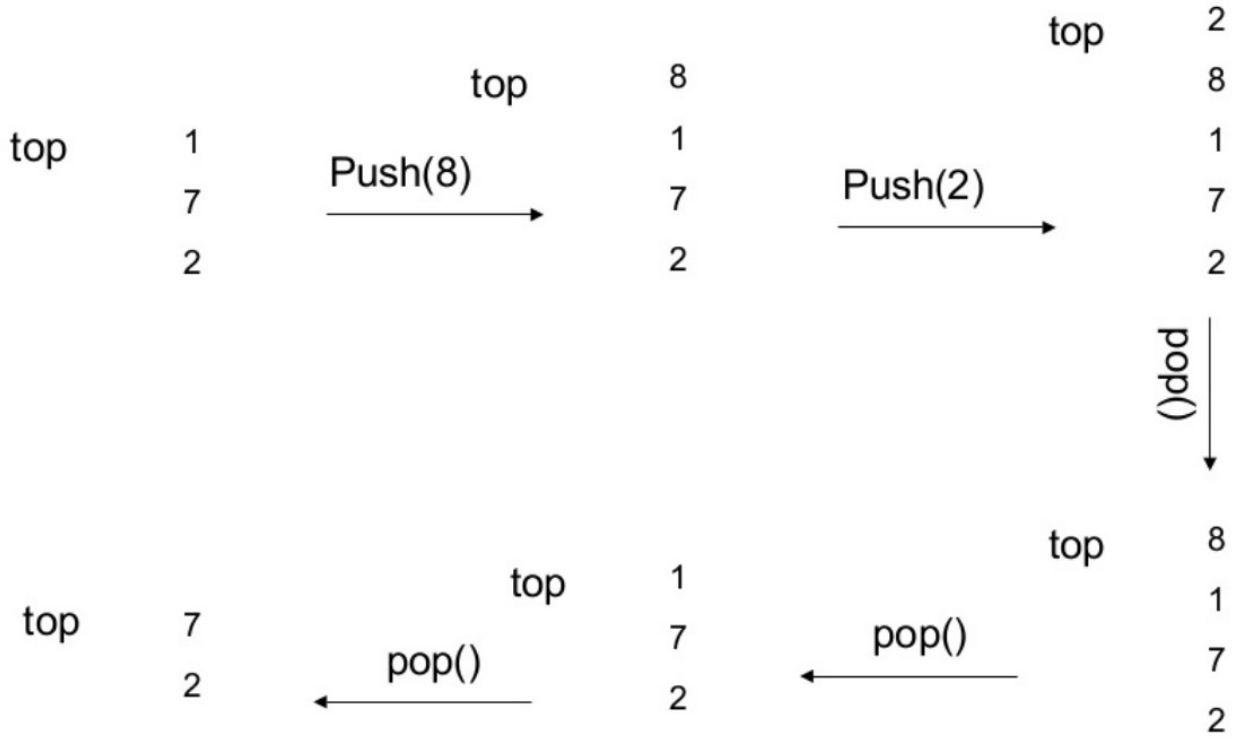
Стек

Що таке стек?

- Стек це структура данних в якій елементи додаються або видаляються тільки з одного кінця, який називається top (вершина)
- Останній елемент який був доданий до стеку буде першим елементом який заберуть зі стеку. Даний принцип має назву LIFO (Last In First Out)
- Додавання елементу дод стеку відбувається за допомогою функції push(), а видалення pop()

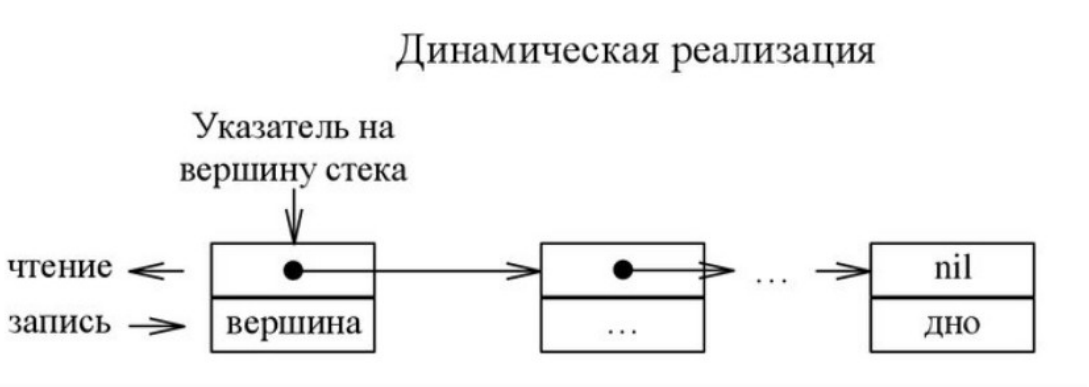
Приклад стеку

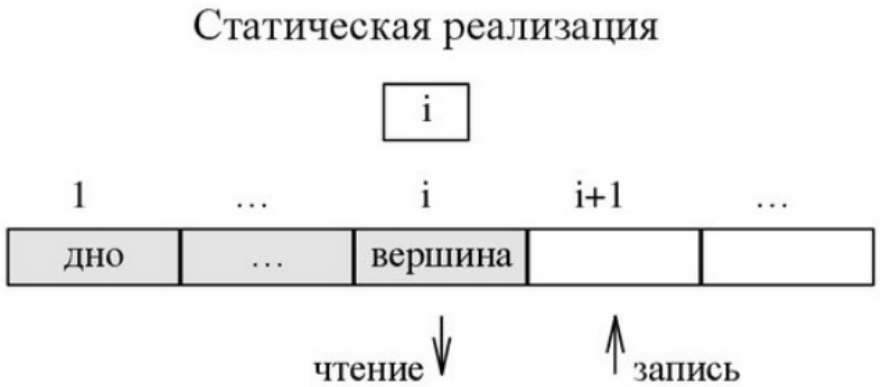


Реалізація стеку

Стеки можуть бути представлені в пам’яті двума способами:

- Статична реалізація – у вигляді вектору (масиву)
- Динамічна реалізація – у вигляд зв’язаного списку





Стек в STL

Функції стеку:

- Push – додати елемент до стеку
- Pop – видалити верхній елемент
- Empty – перевіріти чи стек пустий
- Top – отримати доступ до вершини стека
- Size – отримати розмір стека
- Swap – взаємно змінити вміст двох стеків
- Emplace – додати елеметн до стеку (створення елементу на місці)

```cpp
#include <bits/stdc++.h>
using namespace std;

void showstack(stack <int> s)
{
        while (!s.empty())
        {
                cout << '\t' << s.top();
                s.pop();
        }
        cout << '\n';
}

int main ()
{
        stack <int> s;
        s.push(10);
        s.push(30);
        s.push(20);
        s.push(5);
        s.push(1);

        cout << "The stack is : ";
        showstack(s);

        cout << "\ns.size() : " << s.size();
        cout << "\ns.top() : " << s.top();


        cout << "\ns.pop() : ";
        s.pop();
        showstack(s);

        return 0;
}
```

Області використання

- Передача параметрів в функції
- Трансляція (синтаксичний та семантичний аналізи, генерація коду і т.д)
- Реалізація рекурсії в програмуванні
- Реалізація управління динамічною пам’яттю

Пошук в глибину

- Пошук в глубину – рекурсивний алгоритм пошуку вершин графу або деревоподібної структури.
- Ціль алгоритму обійти всі вершини графу без циклів
- Алгоритм :

1) Вибираємо вершину та помічаємо її як пройдену
2) Додаємо в стек вершини які суміжні до даної
3) Дістаємо елемент зі стеку, помічаємо його як пройдений та додаємо в стек суміжні вершини
4) Повторюємо пункт 3 поки всі вершини не будуть пройденими

- Складність алгоритму по часу O(V + E), по пам’яті O(V)

Пошук в глибину