Олимпиадное программирование
Занятие 4. Контейнеры STL. Итераторы

Труфанов Павел Николаевич

Онлайн-школа  Фоксфорд

Foxford.ru 2019-2020

Vector

Динамический массив.

- Создание - vector<тип>
- Добавление - push_back, insert, emplace_back
- Удаление - pop_back, erase, clear
- Изменение размера - resize, assign
- Размер - size, empty
- Выделение памяти - reserve
- Указатели - begin, end

Stack

Реализация стека

- Создание - stack<тип>
- Вставка - push
- Удаление - pop
- Верхний элемент - top
- Размер - size, empty

Stack с некой функцией от всех его элементов

Хотим уметь поддерживать какую-нибудь функцию для всех текущих элементов в стеке.
Например, минимум.
Будем в элементе хранить две позиции (число, минимум для всех элементов, что находятся под нами).
Тогда легко прописать все операции!

Queue

Реализация очереди

- Создание - queue<тип>
- Вставка - push
- Удаление - pop
- Доступ - front, back
- Размер - size, empty

Задача

Реализовать очередь через стек!
Для того, чтобы сделать очередь, которая умеет поддерживать некоторую функцию от всех элементов в ней!

Queue

Реализация очереди

- Создание - queue<тип>
- Вставка - push
- Удаление - pop
- Доступ - front, back
- Размер - size, empty

Deque

Массив, который умеет вставлять и удалять с обеих сторон за константное время

- Создание - deque<тип>
- Добавление - push_back, push_front, insert
- Еще добавление - emplace_back, emplace_front
- Удаление - pop_back, pop_front, erase, clear
- Изменение размера - resize, assign
- Размер - size, empty
- Указатели - begin, end

Set

set - Реализует отсортированное уникальное множество.
multiset - Реализует отсортированное неуникальное множество.

- Создание set<тип>, multiset<тип>
- Добавление - insert, emplace
- Удаление - clear, erase (можно по указателю и по значению)
- Указатели - begin, end
- Поиск - count, find
- Бинпоиск - lower_bound, upper_bound

Создать set, который сортирует по-твоему

```
bool cmp(const pair<int, int>& a,
         const pair<int, int>& b) {
    return a.first + a.second <
           b.first + b.second;
}

int main() {
    set<pair<int, int>,
        decltype(&cmp)> s(&cmp);
    set<int, greater<int>> s;
}
```

Map

map - Реализует отсортированный уникальный словарь.
multimap - Реализует отсортированный неуникальный словарь.

- Создание map<тип ключа, тип значения>, multimap<тип ключа, тип значения>
- Добавление - insert, emplace
- Удаление - clear, erase
- Указатели - begin, end
- Поиск - count, find
- Бинпоиск - lower_bound, upper_bound

Контейнеры на хеш таблицах

unordered_set
unordered_multiset
unordered_map
unordered_multimap

Итераторы

Вспомним функции begin, end. Что они возвращают?

Например для set<int> они возвращают set<int>::iterator.

Почти все итераторы можно увеличивать. Но лучше всего делать ++it.

Тогда по любой структуре с функциями begin, end можно пройтись двумя способами.

```
for (auto it = s.begin(); it != s.end(); ++it)
for (auto i : s)
```

До встречи!

FOXFORD.RU

Онлайн-школа Фоксфорд

Фоксфорд