

Алгоритмы и структуры данных.

Лектор: Игорь Мамай @igor_mamay

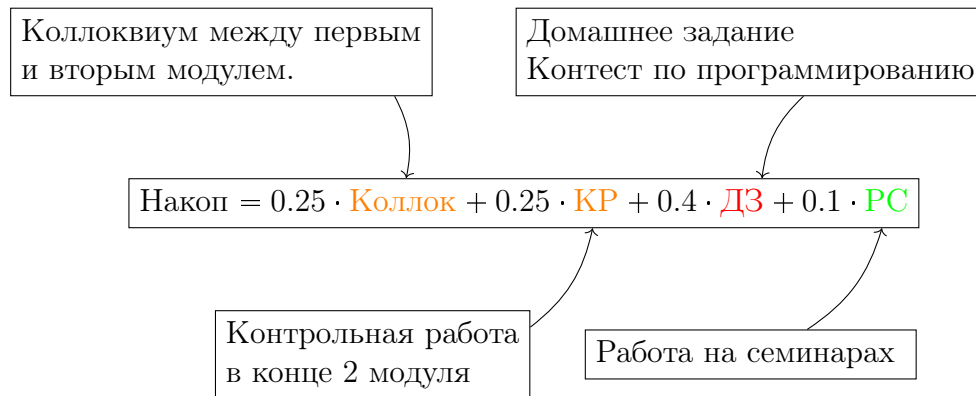
Редактор: Андрей Тищенко @AndrewTGk

2024/2025

Содержание

1	Формула оценки	2
2	Структуры данных	3
2.1	Линейные структуры данных	3
2.1.1	Массив	3
2.1.2	Список	3
2.1.3	Стек	3
2.1.4	Очередь	4
2.1.5	Вектор	4
3	Матчасть для анализа сложности	4
3.1	Метод потенциалов анализа сложности	4
3.2	Символы Ландау	5

1 Формула оценки



2 Структуры данных

Определение

Абстрактный тип данных — объект с определённым набором операций, реализация этих операций умалчивается.

Структура данных — реализация абстрактного типа данных.

2.1 Линейные структуры данных

2.1.1 Массив

Определение

Массив — структура данных, хранящая набор значений (элементов массива) последовательно в памяти.

Реализации массивов в C++

```
typename T arr[20];  
std::array<typename T, 20> arr;
```

2.1.2 Список

Определение

Односвязный список — структура данных, состоящая из элементов одного типа, каждый элемент хранит в себе указатель на следующий элемент.

Двусвязный список — односвязный список, каждый элемент которого дополнительно хранит в себе указатель на предыдущий элемент.

Реализация двух видов списков в C++

```
template<class T>  
struct LinkedList { // односвязный список  
    // Обязательно сюда красоту какую-то напишу  
};  
  
template<typename T>  
struct LinkedListNode {  
    // Обязательно сюда красоту какую-то напишу  
};
```

Замечание

В рамках курса подразумевается, что в программе сохраняется указатель на начало и конец списка.

Элементы списка не обязаны храниться в памяти последовательно.

2.1.3 Стек

Свойства

LIFO (last in, first out)

Различные реализации стека на C++.

```
struct ArrayBasedStack;  
struct ListBasedStack;  
std::deque;
```

Разновидности

Стек с минимумом — сохраняется дополнительный стек, сохраняющий минимумы на префиксах.

2.1.4 Очередь

Свойства

FIFO (first in, first out)

Различные реализации очереди на C++.

```
struct ArrayBasedQueue;  
struct ListBasedQueue;  
struct DequeBasedQueue;  
struct DoubleStackQueue;
```

2.1.5 Вектор

Свойства

Выделяет какое-то базовое количество памяти по умолчанию (зависит от компилятора). Сохраняется указатель на начало, конец используемой пользователем памяти и конец аллоцированной памяти.

Когда конец используемой пользователем памяти совпадает с концом аллоцированной памяти, аллоцируется кусок памяти в 1.5 или 2 (зависит от компилятора) раза больше.

При выполнении n вызовов метода `push_back` вектор перезапишет себя не более $\log n$ раз. В такой ситуации количество перезаписанных элементов α удовлетворяет неравенству:

$$\alpha \leq 1 + \dots + \frac{n}{2} + n \approx 2n = O(n)$$

3 Матчасть для анализа сложности

3.1 Метод потенциалов анализа сложности

Определение

Функцией потенциала можно называть произвольную функцию, которая зависит от параметров структуры данных.

3.2 Символы Ландау

Определение

Функция $f(x)$ называется *оценкой сверху* функции $g(x)$ тогда и только тогда, когда

$$\exists C > 0 \exists x_0 \geq 0 \forall x \geq x_0 : |f(x)| \leq C |g(x)|.$$

Обозначение

$$f(x) = O(g(x))$$

Определение

Функция $f(x)$ называется *оценкой снизу* функции $g(x)$ тогда и только тогда, когда

$$\forall \varepsilon > 0 \exists x_0 \forall x \geq x_0 : |f(x)| \leq \varepsilon |g(x)|.$$

Обозначение

$$f(x) = o(g(x))$$

Определение

Равенство функций $f(x)$ и $g(x)$ выполняется при условии

$$\exists C_1, C_2 \quad (0 < C_1 < C_2) \exists x_0 \forall x \geq x_0 : C_1 |g(x)| \leq |f(x)| \leq C_2 |g(x)|.$$

Обозначение

$$f(x) = \Theta(g(x))$$