

Базовые АД и структуры данных

Гусев Илья, Булгаков Илья

Московский физико-технический институт

Москва, 2018

Содержание

1 АД vs структуры данных

2 Структуры данных

- Куча

АТД vs структуры данных

АТД - абстрактный тип данных.

Набор функций, независимых от конкретной реализации типа, для оперирования его значениями.

Реализация скрыта, по сути представляет из себя интерфейс.

В C++ это абстрактный класс, все функции которого чисто виртуальны (pure virtual).

АТД vs структуры данных

Примеры

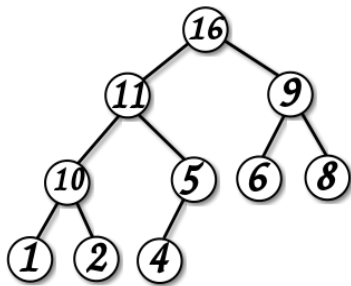
- Стек (push_back, pop_back)
 - ❶ Реализация динамическим массивом
 - ❷ Реализация связным списком
- Очередь (push_back, pop_front)
 - ❶ Реализация динамическим массивом
 - ❷ Реализация связным списком
 - ❸ Реализация двумя стеками
- Очередь с приоритетами (insert(ключ, значение), extract_minimum → (ключ, значение))
 - ❶ Реализация динамическим массивом
 - ❷ Реализация связным списком
 - ❸ Реализация двоичной кучей
 - ❹ Реализация биномиальной кучей
 - ❺ Реализация фибоначчиевой кучей

АТД vs структуры данных

Подробнее про стек и очередь

[External/stacks-algs4.cs.princeton.edu.pdf](https://www.cs.princeton.edu/stacks-algs4.pdf)

Двоичная куча (heap)



- 1 Двоичное дерево (связный ациклический граф, у которого у любой вершины не больше 2 потомков)
- 2 Если узел B является потомком узла A, то $A.key \geq B.key$ (max-куча). Для min-кучи наоборот.
- 3 Глубина всех листьев (расстояние до корня) отличается не более чем на 1 слой.
- 4 Последний слой заполняется слева направо без «дырок».

Двоичная куча (heap, пирамида)

Реализация

16	11	9	10	5	6	8	1	2	4
----	----	---	----	---	---	---	---	---	---

- $A[0]$ - корень
- $\forall i \ A[2i + 1]$ - левый потомок $A[i]$
- $\forall i \ A[2i + 2]$ - правый потомок $A[i]$

Двоичная куча (heap, пирамида)

Действия и сложность

- Добавить элемент в кучу. Сложность $\mathcal{O}(\log n)$
- Исключить максимальный элемент из кучи. Время работы $\mathcal{O}(\log n)$
- Изменить значение любого элемента. Время работы $\mathcal{O}(\log n)$
- Превратить неупорядоченный массив элементов в кучу. Сложность $\mathcal{O}(n)$



Двоичная куча (heap, пирамида)

Построение за $\mathcal{O}(n)$

$\lceil \frac{n}{2^{h+1}} \rceil$ - максимум количества элементов на уровне h

$\mathcal{O}(h)$ - сложность вставки элемента на уровень h

$\lfloor \lg(n) \rfloor$ - высота n -элементной пирамиды

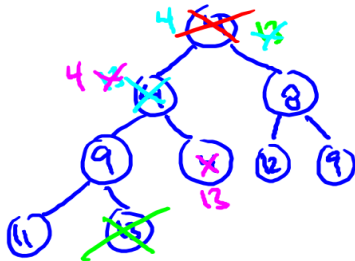
$$\sum_{h=0}^{\lfloor \lg(n) \rfloor} \lceil \frac{n}{2^{h+1}} \rceil \mathcal{O}(h) = \mathcal{O}(n \sum_{h=0}^{\lfloor \lg(n) \rfloor} \frac{h}{2^h})$$

$$\sum_{h=0}^{\infty} \frac{h}{2^h} = \frac{1/2}{(1 - 1/2)^2} = 2$$

$$\mathcal{O}(n \sum_{h=0}^{\lfloor \lg(n) \rfloor} \frac{h}{2^h}) = \mathcal{O}(n \sum_{h=0}^{\infty} \frac{h}{2^h}) = \mathcal{O}(2n) = \mathcal{O}(n)$$

Двоичная куча (heap, пирамида)

Вставка и extract_min



Полезные ссылки I



Т.Кормен, Ч.Лейзерсон, Р.Ривест, К.Штайн - Алгоритмы.
Построение и анализ. Глава 6

<https://bit.ly/2wFzphU>



Lecture Slides for Algorithm Design

<https://algs4.cs.princeton.edu/lectures/>