Сливаемые кучи: биномиальная куча, куча Фибоначчи

Булгаков Илья, Валерий Сенотов

Московский физико-технический институт

Москва, 2023

Содержание

Сливаемые кучи

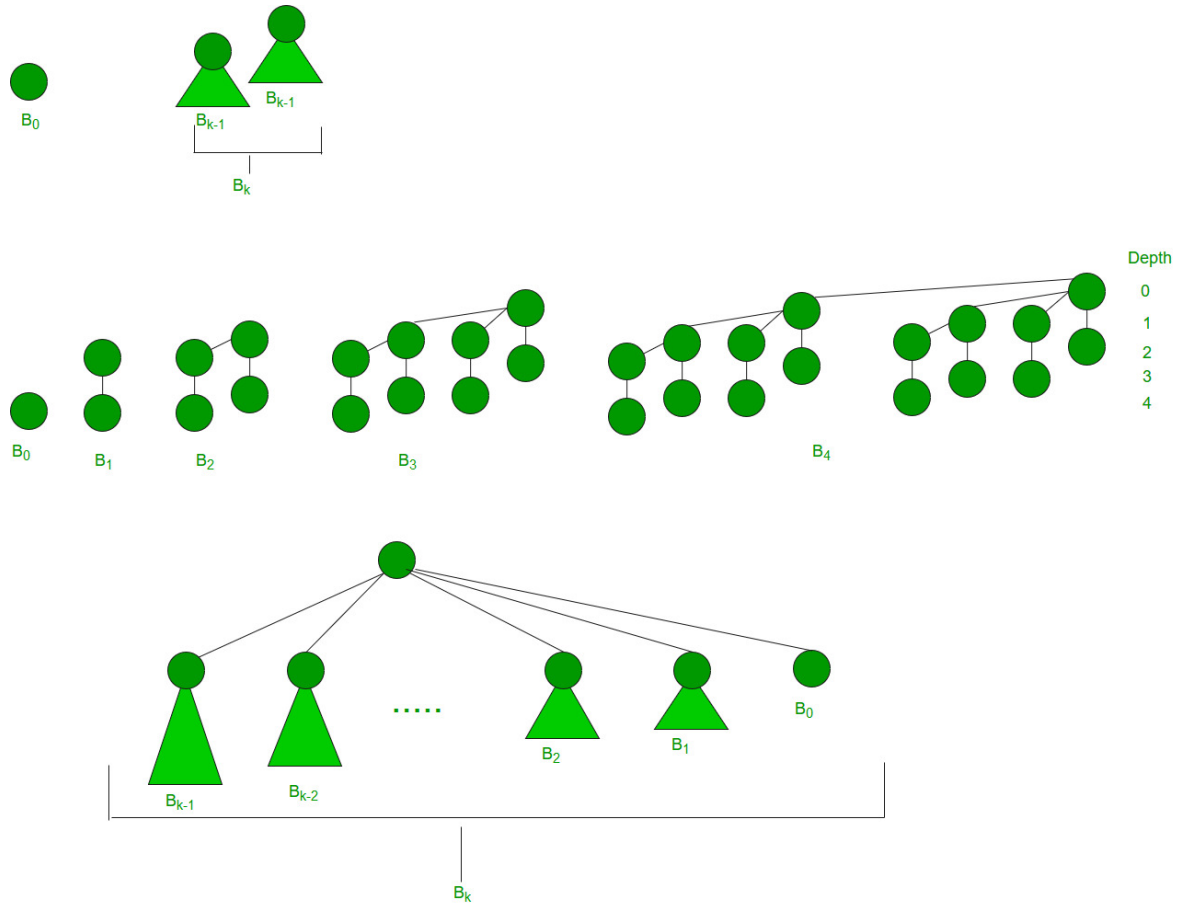
Сливаемые кучи – класс структур данных, которые реализуют операцию Merge (Union), то есть создание новой кучи H, которая содержит все узлы куч H_1 и H_2
Операции:

- Insert
- GetMin
- ExtractMin
- **Merge**

Биномиальное дерево

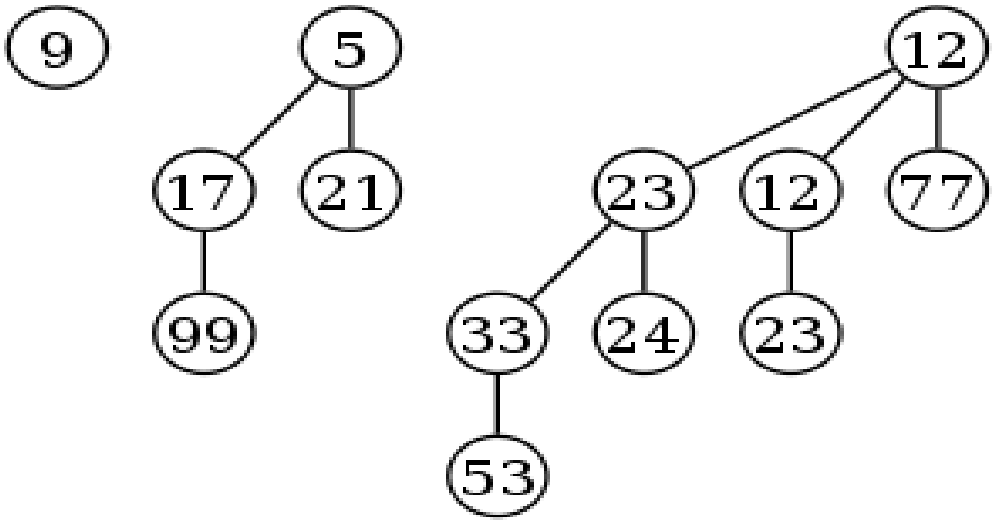
Определение

Биномиальное дерево B_k состоит из двух биномиальных деревьев $B_(k-1)$, связанных вместе таким образом, что корень одного из них является дочерним узлом корня второго дерева. B_k – дерево из одного узла.



Биномиальная куча

Определение



Биномиальная куча:

1. является объединением биномиальных деревьев;
2. каждое биномиальное дерево B_k в куче подчиняется свойству неубывающей кучи;
3. $\forall k$ в куче существует не более одного B_k.

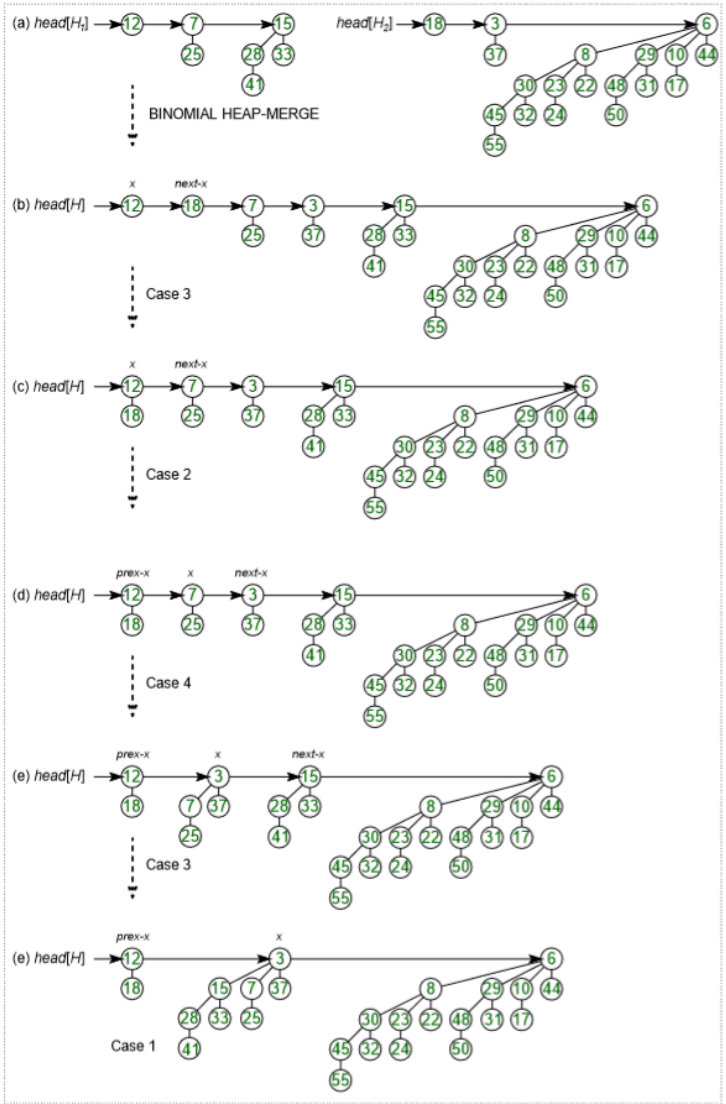
Биномиальная куча

Операции

- Insert
- GetMin
- ExtractMin
- Merge

Биномиальная куча

Merge



- Добавляем в кучу H_1 по порядку деревья из кучи H_2.
- Если добавляем дерево высоты k:
  1. В H_1 нет дерева такой высоты, то просто добавляем это дерево в множество кучи H_1.
  2. В H_1 есть дерево такой высоты, то удаляем это дерево из H_1 и соединяем эти деревья в дерево высоты $k+1$ с сохранением свойства кучи. Добавляем получившееся дерево в H_1.

Биномиальная куча

Операции

- Insert: слить данную кучу и кучу из одного элемента.
- GetMin: найти минимум среди корней деревьев.
- ExtractMin: удалить дерево с минимумом, слить данную кучу с кучей, полученной после удаления минимума из дерева.

Сложность

- Insert: $O(logn)$
- GetMin: $O(logn)$
- ExtractMin: $\Theta(logn)$
- Merge: $O(logn)$

Биномиальная куча

Реализация

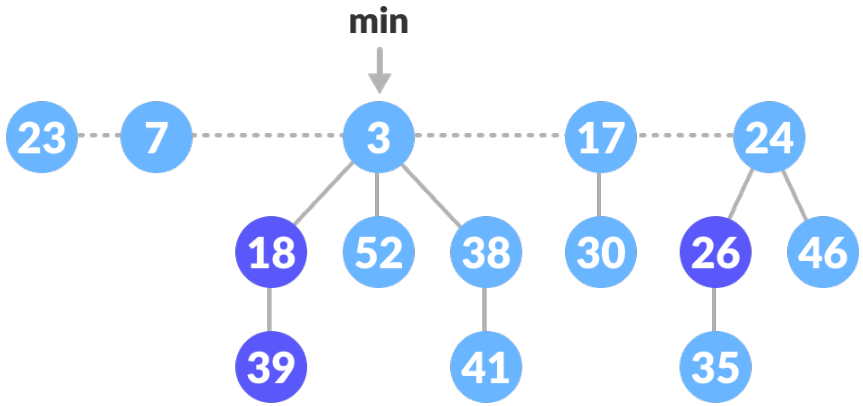
```
struct Node {
    int key;
    Node* parent;
    Node* child_left;
    Node* child_right;
    int degree;
}

struct BinomHeap {
    std::vector<Node> head;
}
```

Куча Фибоначчи

Определения

Фибоначчиева куча — набор из подвешенных деревьев, удовлетворяющих свойству кучи, поддерживающий указатель на минимум элементов кучи.
Степень вершины — количество детей данной вершины.
Степень кучи — наибольшая степень вершины этой кучи.



Куча Фибоначчи

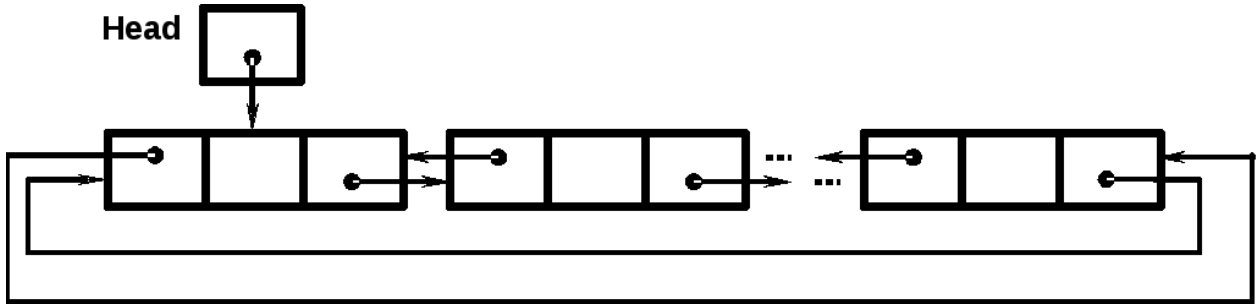
Операции

- Insert
- GetMin
- ExtractMin
- Merge

Куча Фибоначчи

Операции

- Insert: добавить элемент как новое дерево, обновить указатель на минимум, если нужно ($O(1)$).
- GetMin: передать значение по указателю ($O(1)$).
- Merge: храним деревья в циклическом двусвязном списке; чтобы объединить кучи, соединить списки деревьев, обновить указатель на минимум, если нужно ($O(1)$).



Куча Фибоначчи

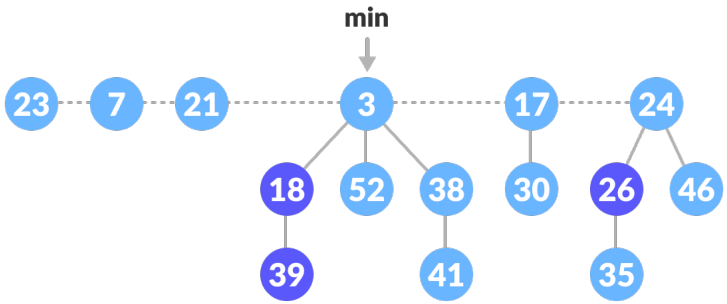
ExtractMin

1. удалить минимум по указателю
2. добавить поддеревья минимума в кучу
3. перенести указатель на корень следующего дерева
4. создать массив длиной, равной степени кучи + 1
5. соотнести ячейки массива по номеру с указателями на деревья соответствующей степени до первого совпадения
6. объединить деревья со совпадающей степенью
7. объединение заканчивается, когда степень всех деревьев попарно различна.

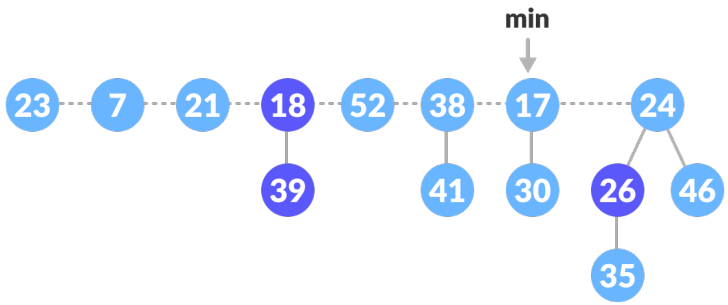
Куча Фибоначчи

ExtractMin

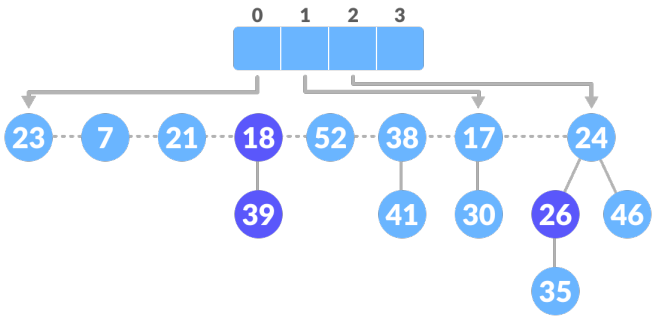
1. Удаляем минимум.



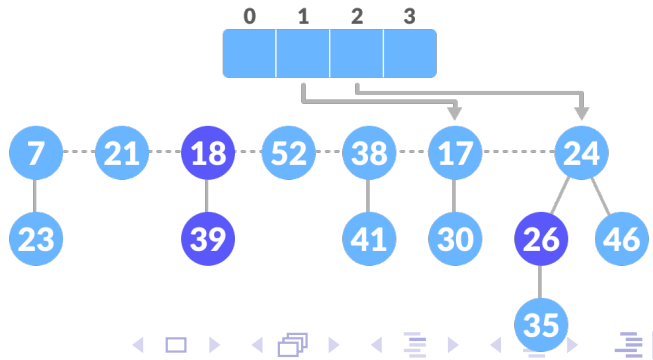
2-3. Добавляем поддеревья в кучу.



4-5. Создаем массив для вершин деревьев с разной степенью.



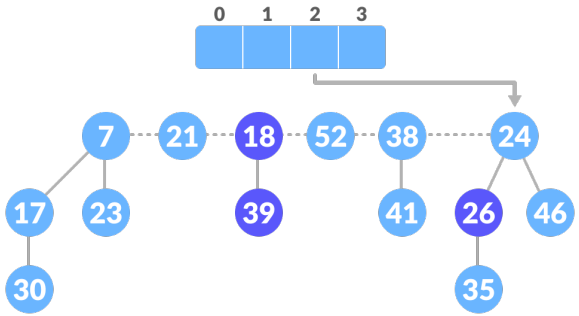
6. Объединяем деревья по списку, начиная с дерева [23].



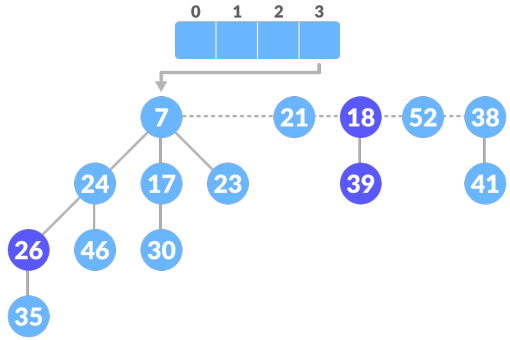
Куча Фибоначчи

ExtractMin

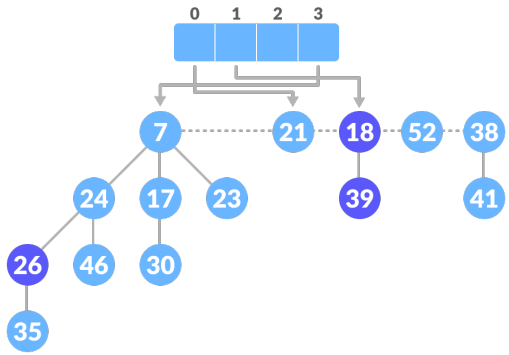
6. Объединяем деревья степени 1.



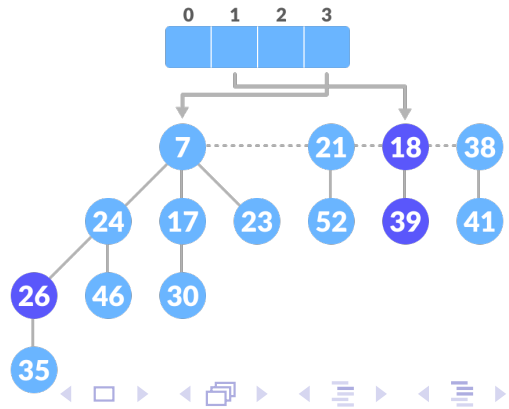
6. Объединяем деревья степени 2.



6. Добавляем новые деревья в массив.



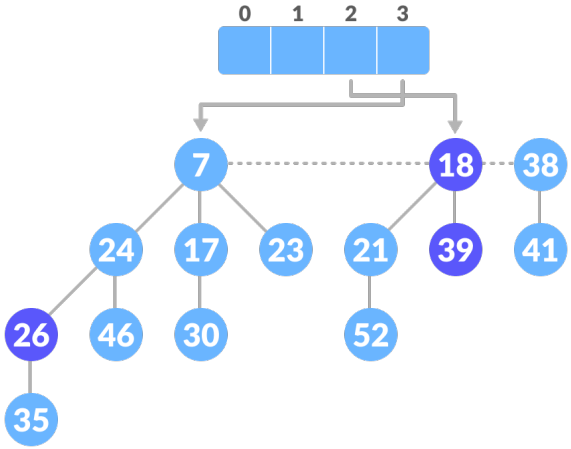
6. Объединяем деревья степени 0.



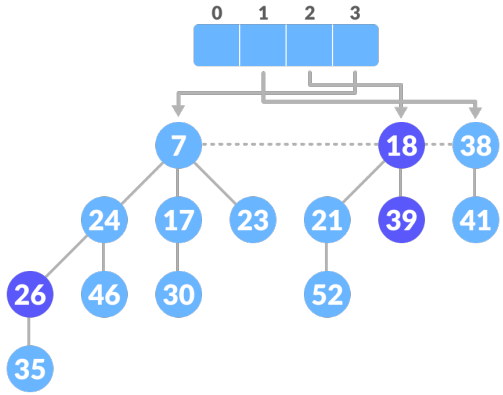
Куча Фибоначчи

ExtractMin

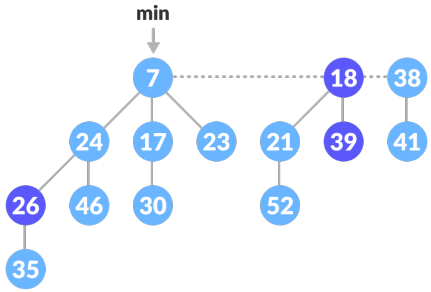
6. Объединяем деревья степени 1.



6. Добавляем новые деревья в массив.



7. Заканчиваем объединение.



Куча Фибоначчи

Амортизированная сложность

- Insert: O(1)
- GetMin: O(1)
- ExtractMin: O(logn)
- Merge: O(1)

DecreaseKey

Какие операции (за $O(logn)$) мы можем производить с кучей?

- Строить кучу
- Добавлять элементы в кучу
- Читать и удалять минимум кучи

Какие операции (за $O(logn)$) еще могли бы быть полезны?

- Изменять произвольный элемент
- Удалять произвольный элемент

DecreaseKey

Бинарная куча

DecreaseKey(elem, sub) — уменьшение элемента *elem* в куче на значение *sub*
DecreaseKey(*elem*, *sub*):

1. Find(elem)
2. UpdateValue(elem, sub)
3. SiftUp(elem)

DecreaseKey

Бинарная куча

Для эффективного Find(elem) добавим в структуры кучи map.
$map(key_{elem}) \longrightarrow index(elem)$
Например, map можно реализовать с помощью хэш-таблицы, ключом будет значение элемента.

```
void Swap(unsigned int i, unsigned int j) {
    int temp = elems[i];
    elems[i] = elems[j];
    elems[j] = temp;
    map[elems[i]] = i;
    map[elems[j]] = j;
}

void DecreaseKey(int elem, int sub) {
    unsigned int i = map[elem];
    elems[i] -= sub;
    SiftUp(i);
}
```

DecreaseKey

Биномиальная куча, куча Фибоначчи

Биномиальная куча:
$map(key_{elem}) \longrightarrow pointer(elem)$
Куча Фибоначчи:
$map(key_{elem}) \longrightarrow pointer(elem)$

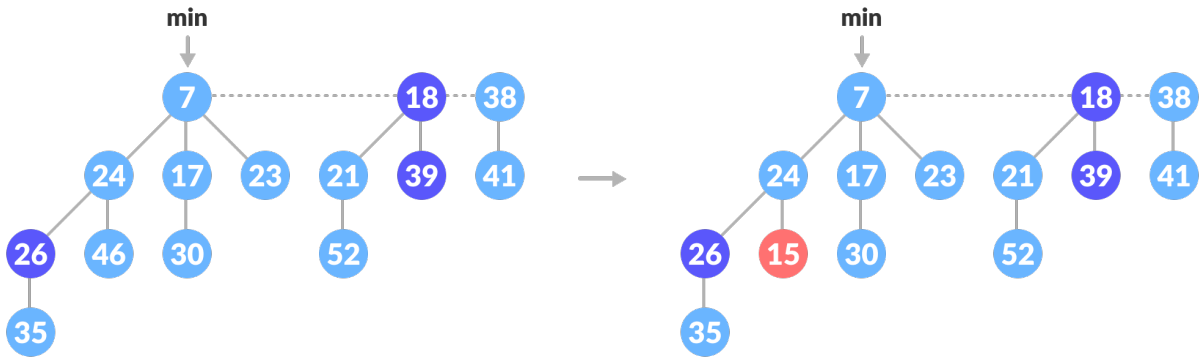
```
struct Node{
    int value;
    Node* parent;
    Node* left;
    Node* right;
    Node* left_child;
    bool to_cut;
}
```

```
void DecreaseKey(elem, sub) {
    Node* pNode = Find(elem);
    Node* parent = pNode->parent;
    void Cut(Node* ptr) {
        ptr->left->right = ptr->right;
        ptr->right->left = ptr->left;
        ptr->parent = ptr->left = ptr->
            right = nullptr;
        ptr->to_cut = false;
    }
    Cut(pNode);
    Insert(pNode);
    while (parent->to_cut)
        Cut(parent);
    parent->to_cut = true;
    UpdateMin();
}
```

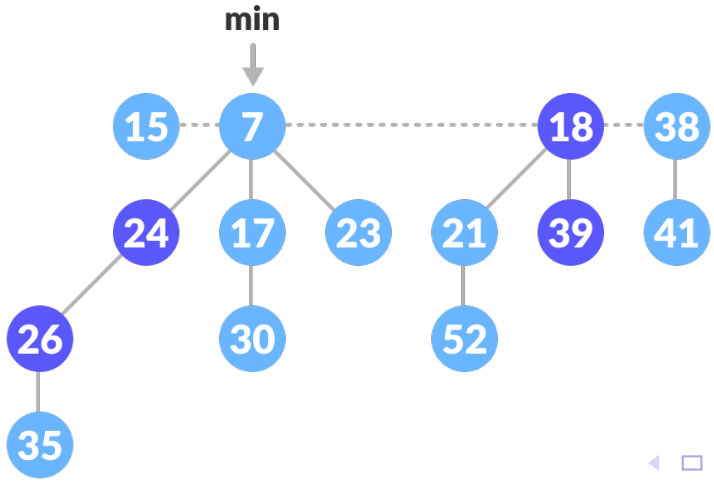
DecreaseKey

Куча Фибоначчи

DecreaseKey(45, 15):



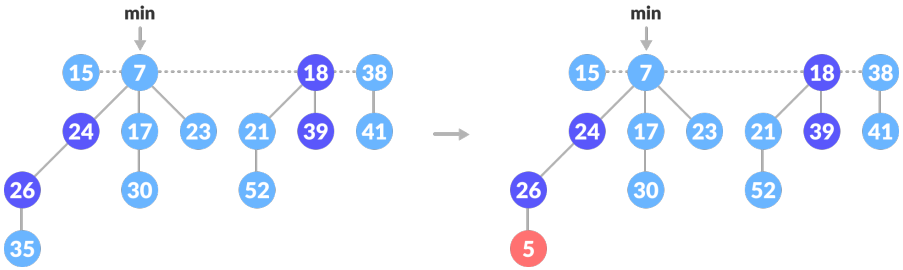
Cut(15)



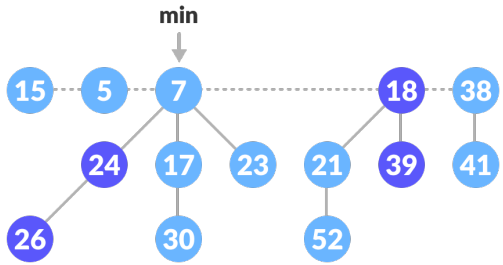
DecreaseKey

Куча Фибоначчи

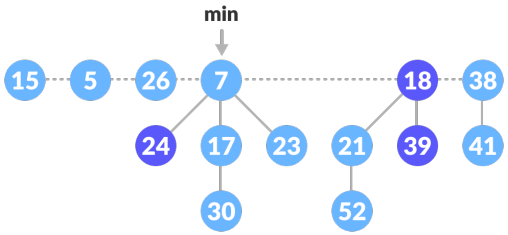
DecreaseKey(35, 5):



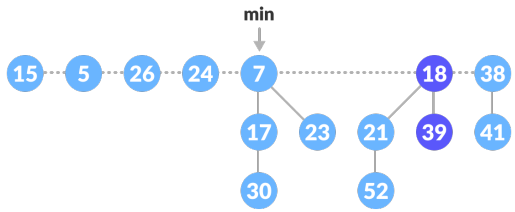
Cut(5)



Cut(26)



Cut(24)



DecreaseKey

Биномиальная куча, куча Фибоначчи

Как реализовать Delete?
Как реализовать IncreaseKey?

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/


Implementing the Decrease-Key
https://www.baeldung.com/cs/min-heaps-decrease-key


Хабр. Кучи. Часть 1. Биномиальная куча
https://habr.com/ru/articles/135232/