

**Санкт-Петербургский
политехнический университет имени
Петра Великого**
Кафедра “Прикладная математика и информатика”

Отчет по Летней практике
“Алгоритмы и структуры данных”

Студент группы № 5030102/20001 _____ Соколов Артем Николаевич

Подпись руководителя _____ Козлов Константин Николаевич

Санкт-Петербург

2024 год

Оглавление

Титульный лист	
Постановка задачи	3
Описание алгоритма	3
Структура кучи	3
Добавление элемента в кучу	3
Объединение двух куч.	3
Удаление минимального элемента.	4
Текст программы	4
Описание тестирования	4
Тест 1. Добавление элемента в кучу	5
Тест 2. Удаление минимального элемента из кучи	6
Тест 3. Объединение двух куч.	9

Постановка задачи

В работе требуется реализовать структуру данных фибоначчиева куча, а также операции добавления элемента, объединения двух куч и удаления минимального элемента.

Пользователь программы может делать следующие запросы:

- Добавить элемент в кучу.
- Объединить две кучи.
- Удалить минимальный элемент из кучи.

Описание алгоритма

Структура кучи

1. Фибоначчиева куча – это куча (список деревьев), на каждое дерево которого, помимо ограничения, что значение в родителе (для min) должно быть меньше значения потомков, также присутствует ещё одно: для родителя ранга k его потомки слева направо должны быть ранга не меньше, чем от 0 до $k - 1$. Также в данной куче отдельно хранится минимальный элемент.

Добавление элемента в кучу

Элемент добавляется как новое дерево в кучу и, если он меньше минимального элемента, минимальный элемент заменяется на добавленный. Сложность $O(1)$.

Объединение двух куч.

Объединяются два списка деревьев: один объединяется с концом другого. Сложность $O(1)$.

Удаление минимального элемента.

Здесь выполняются 2 действия: удаление элемента и поиск нового с одновременным сжатием кучи.

1. Удаление элемента: элемент удаляется из списка по хранящейся отдельно ссылке, его потомки переносятся в корневой список деревьев.
2. Сжатие с поиском нового минимума:
 - a. Заводим массив, в котором индексом будет являться ранг дерева, а значением – ссылка на корень дерева этого ранга.
 - b. Обходим список деревьев слева-направо, с поиском минимума.
 - c. Проверяем, что находится в массиве по рангу текущего дерева. Если там ничего нет, что записываем туда текущее дерево. Если же там уже есть дерево такого же ранга, то «подвешиваем» одно дерево к другому (в зависимости от того, у кого меньше корневой элемент), увеличивая ранг итогового дерева на 1.
 - d. Далее повторяем пункт c. для получаемого дерева до того момента, пока в массиве не окажется пустого места.

Амортизированная сложность удаления равна $O(\log(n))$

Исходные файлы программы

Исходный код проекта: [azyua0/FibonacciHeap \(github.com\)](https://github.com/azyua0/FibonacciHeap)

Весь код Фибоначчиевой кучи был реализован в .h файлах из-за использования технологии “Шаблоны”.

В main.cpp файле записан код для тестирования.

Тестирование

```
5  ∨ int main() {  
6      auto data1 = FibonacciHeap<float>();  
7      data1.insert(1);  
8  
9      data1.insert(2);  
10     data1.insert(3);  
11     data1.insert(0.5f);  
12  
13     auto data2 = FibonacciHeap<float>();  
14     data2.insert(1234);  
15  
16     data2.insert(232);  
17     data2.insert(43);  
18     data2.insert(0.25f);  
19  
20     FibonacciHeap<float>::join(&data1, &data2);  
21  
22     data1.printTop();  
23     std::cout << '\n' << data1.getMin() << '\n';  
24  
25     data1.compress();  
26  
27     data1.printTop();  
28  
29     std::cout << '\n' << data1.popMin() << '\n';  
30     std::cout << data1.getSize() << '\n';  
31  
32     return 0;  
33 }  
34
```

Рис 1. Исходные команды для тестирования

```

C:\Users\Azya\Desktop\code\university\SummerTask\build>ninja
[2/2] Linking CXX executable SummerTask.exe

C:\Users\Azya\Desktop\code\university\SummerTask\build>.\SummerTask.exe
0.25 0.5 3 2 1 1234 232 43
0.25
0.25
0.25
7

```

Рис 2. Данные вывода для тестов

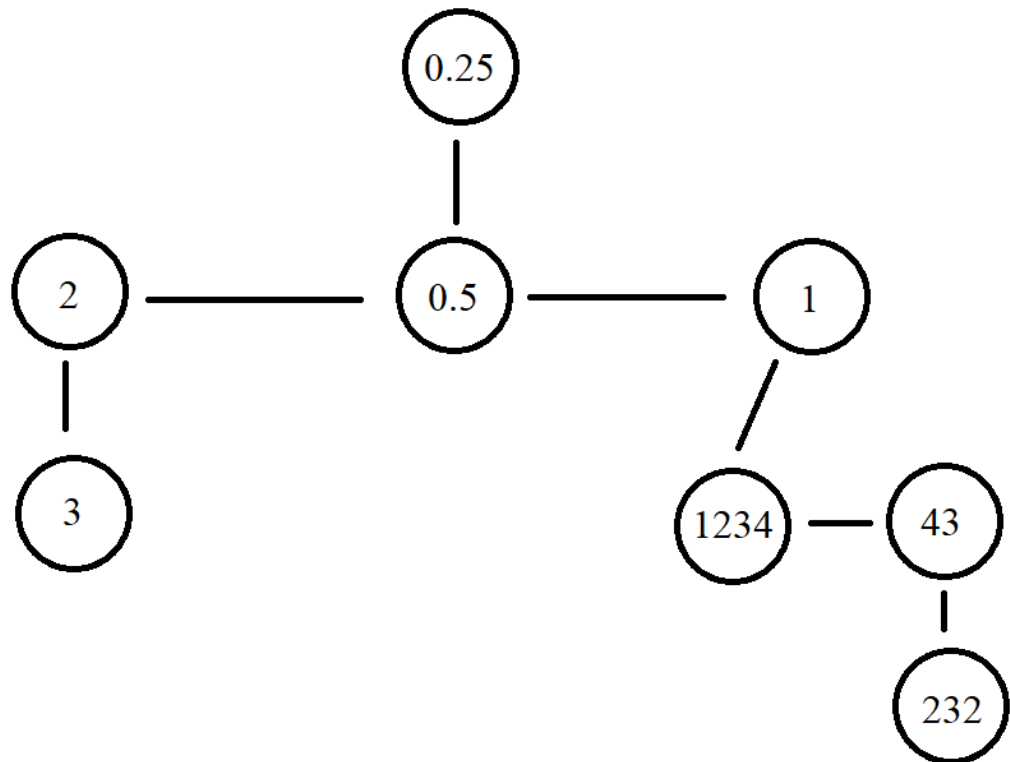


Рис 3. Структура Фибоначчиевой кучи после первого уплотнения

Итоги

В рамках данной работы я реализовал операции “Добавить элемент”, “Получить минимальный элемент”, “Удалить минимальный элемент”, “Объединить две кучи” для структуры данных “Фибоначчиева куча”.