**Киевский национальный университет имени Тараса Шевченка**

**Факультет компьютерных наук и кибернетики**

**Алгоритмы и сложность**

**Задание № 7**

**Биноминальная пирамида**

**Отчет**

**Выполнила:**

студентка группы К-29

Кошель Анастасия Андреевна

**Киев-2018**

1. **Условие задания.** Реализовать биноминальную пирамиду.
2. **Алгоритм.**

Для описание пирамиды используем класс BinHeap, который содержит структуру node с полями data, degree(количество дочерних узлов данного узла), parent, sibling, child;

Основные функции:

*insert*(вставка элемента, по факту обьединяет имеющуюся кучу с кучей состоящей из одного элемента, который и нужно добавить; время работы О(logn))

*getmin* (возвращает элемент с наименьшим значением, корень каждого дерева по определению является минимальным из всех элементов данного дерева, поэтому достаточно найти минимум в списке корней; время работы также О(logn))

*extractmin* (извлекает узел с минимальным ключом из биномиальной кучи и возвращает указатель на извлеченный узел)

*merge* (соединяет две биноминальные кучи в одну; сначала соединяем два корневых списка куч в один, потом вызываем функцию fix и join, которая обьединяет 2 дерева одинакового размера в одно; время работы Ω(logn))

*decreasekey* (уменьшает ключ элемента x биномиальной кучи, присваивая ему новое значение; Вершина, ключ которой был уменьшен, «всплывает» как в обычной куче; время Θ(logn) )

1. **Вывод.**

**Биномиальная куча** представляет собой множество биномиальных деревьев, которые удовлетворяют следующим свойствам:

* каждое биномиальное дерево в куче подчиняется свойству **неубывающей кучи**: ключ узла не меньше ключа его родительского узла (упорядоченное в соответствии со свойством неубывающей кучи дерево),
* для любого неотрицательного целого kk найдется не более одного биномиального дерева, чей корень имеет степень kk.

1. **Использованные источники.**

neerc.ifmo.ru