Глава 1.Тонкая куча(thin heap)

Тонкая куча – это структура данных похожая на фибоначчиеву кучу, и имеющая такие же асимптотические оценки. Время работы операций create, insert, find-min, merge, decreaseKey – $O(1)$, а операций delete, delete-min – $O(\log n)$. Тонкая куча, при одинаковых асимптотических оценках, имеет меньшее кол-во констант по сравнению с фибоначчиевой кучей, что на практике должно ускорить работу и уменьшить объем затраченной памяти.

Тонкое дерево – это биномиальное дерево, каждый узел которого мог потерять левого ребенка(или поддерево узлом которого являлся левый ребенок).

Более формально тонкое дерево – это упорядоченное дерево, каждый узел которого имеет положительный или равный 0 ранг, и удовлетворяет следующим свойствам:

1. узел с рангом r, либо имеет r детей с рангами r - 1, r - 2, …, 0 (такой узел называют толстым), либо имеет r - 1 детей с рангами r -2, r - 3, …, 0 (такой узел называют тонким);

2. корень является толстым узлом.

Заметим, что если соединить два тонких дерева, корни которых имеют одинаковый ранг r, сделав один из корней левым ребенком другого, то получится тонкое дерево с корнем ранга r + 1.

Тонкая куча – это набор кучеобразных(т.е. удовлетворяющих условиям кучи) тонких деревьев

Структура узла и кучи:

```
template <class T>
class Node
{
        T key; //ключ
        int rank; //ранг
        Node* child; //указатель на самого левого ребенка
        Node* right; //указатель на правого брата, либо на следующий корень, если
    текущий узел является корнем
        Node* left; // указатель на левого брата , либо на родителя, если текущий узел
    самый левый, либо null, если текущий узел корень
};

template <class T>
class thinheap
{
        Node* first; //указатель на корень с минимальным ключем
        Node* last; //указатель на последний корень
};
```