**Киевский национальный университет имени Тараса Шевченка**

**Факультет компьютерных наук и кибернетики**

**Алгоритмы и сложность**

**Задание № 3**

**Splay Tree**

**Отчет**

**Выполнила:**

студентка группы К-29

Кошель Анастасия Андреевна

**Киев-2018**

**Splay Tree**

1. **Условие задания.** Реализовать splay tree.
2. **Алгоритм.**

Для описания дерева используем структуру node, которая содержит поле с данными и указатели на правого и левого сына, а также на родителя.

Основная эвристика splay-дерева — move-to-root. После обращения к любой вершине, она поднимается в корень. Подъем реализуется через повороты вершин. За один поворот, можно поменять местами родителя с ребенком (right\_rotate, left\_rotate).

Но просто поворачивать вершину, пока она не станет корнем, недостаточно. Хитрость splay-дерева в том, что при продвижении вершины вверх, расстояние до корня сокращается не только для поднимаемой вершины, но и для всех ее потомков в текущих поддеревьях. Для этого используется техника zig-zig и zig-zag поворотов.  
  
Основная идея zig-zig и zig-zag поворотов, рассмотреть путь от дедушки к ребенку. Если путь идет только по левым детям или только по правым, то такая ситуация называется zig-zig. В противном случае, мы сначала меняем ребенка с текущим родителем, потом с новым. Если у вершины дедушки нет, делаем обычный поворот. В программе это выполняет функция splay.

Процедура поиска в splay-дереве отличается от обычной только на последней стадии: после того, как вершина найдена, мы тянем ее вверх и делаем корнем через процедуру splay.

Вставка ключа происходит также как и в обычном бинарном дереве поиска, только в конце вызывается функция splay.

Для того чтобы удалить вершину поднимаем ее наверх при помощи функции splay, а затем, если левое поддерево пустое делаем верхний элемент правого корнем и завершаем работу. Если левое поддерево непустое, то находим в нем «самый правый элемент» и делаем его корнем.

**Анализ алгоритма.**

Процедуры удаления, вставки, слияния и разделения деревьев работают за http://habrastorage.org/getpro/habr/post_images/ec7/176/ecf/ec7176ecffb2f187d5ef0d0d24ffdbd4.gif +время работы процедуры find.  
Процедура find работает пропорционально глубине искомой вершины в дереве. По завершении поиска запускается процедура splay, которая тоже работает пропорционально глубине вершины. Таким образом, достаточно оценить время работы процедуры splay.

Операция splay  выполняется за http://habrastorage.org/getpro/habr/post_images/84c/07b/cc9/84c07bcc99d5fc8ab9086ace521ed96a.gif.

1. **Вывод.**

Splay-дерево — это самобалансирующееся бинарное дерево поиска. Дереву не нужно хранить никакой дополнительной информации, что делает его эффективным по памяти. После каждого обращения, даже поиска, splay-дерево меняет свою структуру. .

1. **Использованные источники.**

Habrahabr.ru