

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**

**по курсовой работе**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: «Рандомизированная пирамида поиска. Текущий контроль.»**

Студент гр. 8381

Сосновский

Д.Н.

Преподаватель

Жангиров Т.Р.

Вариант 15  
Санкт-Петербург  
2019

## **Цель работы**

Детально ознакомиться со структурой данных рандомизированной пирамиды, реализовать опцию «Текущий контроль», позволяющую преподавателю генерировать задания с ответами с использованием данной структуры данных.

## **Ход работы**

Для удобного и приятного взаимодействия с программой были добавлены некоторые вспомогательные вещи: ползунки в области помещения рисунка, различные Actions, при нажатии на некоторые из которых всплывает окошко со вспомогательной информацией. В одном из Actions была помещена опция текущего контроля. При нажатии на этот Action появляется ещё одно окно, в котором пользователю предлагается выбрать файл для записи в него задания. Файл должен иметь расширение .txt, поэтому диалог выбора файла откроется сразу же с настроенной маской поиска. Следует отметить, что при попытке сгенерировать задание в несуществующий файл, т.е. которого нет по заданному пути (после выбора файла его путь отобразится в небольшом пространстве под кнопкой) в поле имени файла появится надпись «Файл не существует», что говорит о неправильном задании адреса файла. Таким образом, программа не позволит записать в неверный файл информацию. Если же файл успешно найден и открыт, то тогда программа сгенерирует в него задание с пошаговым решением. Само задание состоит из двух пунктов: добавление вершин в рандомизированную пирамиду поиска, т.е. построение рандомизированной пирамиды по последовательности вершин, и удаление вершин из построенной пирамиды. Второе задание логически предполагает выполнение первого, так как в нём используется построенная на первом этапе рандомизированная пирамида поиска. В файле, помимо самого

задания, будет так же выводиться и пошаговое решение данной задачи. Для этого на каждой новой строчке будет написано состояние дерева в обходе в ширину (в такой записи символ «#» означает отсутствие очередного корня, т.е. пустое дерево). При желании можно проверить выполнение работы с помощью разработанной для предыдущего задания системы генерации рандомизированной пирамиды поиска на основе последовательности из значений с приоритетами. Для этого необходимо переключиться на первое окно, которое появлялось при запуске программы, и вставить в окошко ввода ту последовательность, которая была сгенерирована в файле. Таким образом осуществляется проверка работы разработанного алгоритма.

### **Оценка сложности работы программы**

Данная программа осуществляет генерацию рандомизированной пирамиды поиска за время  $O(n \log_2 n)$ . Отрисовка рандомизированной пирамиды поиска осуществляется за время  $O(n)$ . Генерация заданий осуществляется за время  $O(n \log_2 n)$ , исключая слагаемые меньшего порядка из скобок. Графики оценки времени выполнения приведены в приложении.

### **Тестирование и пример работы программ**

Поставим сложность генерирования задачи равной 5. Содержимое файла с задачей приведено в приложении в листинге 1.

Повторим опыты с сложностями 9, 13, 14. Содержимые файлов с задачами приведены в листингах 2, 3 и 4 соответственно.

### **Выводы**

В ходе выполнения данной работы детально была изучена структура данных рандомизированной пирамиды. Была разработана программа, позволяющая преподавателю генерировать задачи с готовыми пошаговыми

ответами для удобной проверки. Задания готовы к передаче преподавателю и проведению семинаров на тему построения рандомизированной пирамиды поиска.

## **ЛИТЕРАТУРА**

- 1) Статья «Алгоритмы и структуры данных» <https://ppt-online.org/672078>  
(Обращение происходило в период с 19.12.19 по 26.12.19)
- 2) Онлайн – платформа для обучения Stepik (обращение происходило в период с 24.12.19 по 26.12.19)
- 3) Статья на Хабрахабре (обращение в период с 19.12.19 по 26.12.19)  
<https://habr.com/ru/post/145388/>

## ПРИЛОЖЕНИЕ

### Листинг 1.

Для заданного входного набора ключей и приоритетов построить последовательность рандомизированных пирамид поиска, полученных поочередной вставкой данных их входного набора.

Входной набор:

(91 ; 1), (20 ; 5), (52 ; 2), (57 ; 2), (59 ; 4)

Шаг 1: 91

Шаг 2: 20#91

Шаг 3: 20#52###91

Шаг 4: 20#52###57#####91

Шаг 5: 20#59##5291#####57

Для полученного дерева исключить элемент 57

Шаг 1: 20#59##5291

### Листинг 2.

Для заданного входного набора ключей и приоритетов построить последовательность рандомизированных пирамид поиска, полученных поочередной вставкой данных их входного набора.

Входной набор:

(100 ; 1), (3 ; 6), (82 ; 3), (82 ; 6), (100 ; 3), (62 ; 2), (50 ; 7), (10 ; 2), (14 ; 2)

Шаг 1: 100

Шаг 2: 3#100

Шаг 3: 3#82###100

Шаг 4: 3#82##82100

Шаг 5: 3#82##82100#####100

Шаг 6: 3#82##82100####62#100

Шаг 7: 50382##82100####62#100

Шаг 8: 50382#1082100####62#100

Шаг 9: 50382#1082100###1462#100

Для полученного дерева исключить элемент 10

Шаг 1: 50382#1482100##10#62#100

Шаг 2: 50382#1482100####62#100

### **Листинг 3**

Для заданного входного набора ключей и приоритетов построить последовательность рандомизированных пирамид поиска, полученных поочередной вставкой данных их входного набора.

Входной набор:

(1 ; 8), (24 ; 6), (16 ; 11), (81 ; 9), (40 ; 10), (86 ; 7), (32 ; 4), (69 ; 8), (85 ; 1), (8 ; 9), (57 ; 11), (45 ; 7), (25 ; 3)

Шаг 1: 1

Шаг 2: 1#24

Шаг 3: 16124

Шаг 4: 16181##24

Шаг 5: 16140##2481

Шаг 6: 16140##2481#####86

Шаг 7: 16140##2481#####32#86

Шаг 8: 16140##2481#####326986

Шаг 9: 16140##2481#####326986#####85

Шаг 10: 168401#2481#####326986#####85

Шаг 11: 168571#4081#####24#6986#####32####85

Шаг 12: 168571#4081#####24456986#####32####85

Шаг 13:

168571#4081####24456986#####32####85#####25

Для полученного дерева исключить элемент 16

Шаг 1: 57168184069861#2445##85#####32#####25

Шаг 2: 57408116456986824####85#1##32#####25

Шаг 3:

5740818456986116####85####24#####32#####  
#####25

Шаг 4: 5740818456986124####85###1632#####25

Шаг 5: 5740818456986124####85####32#####25

#### Листинг 4

Для заданного входного набора ключей и приоритетов построить последовательность рандомизированных пирамид поиска,

полученных поочередной вставкой данных их входного набора.

Входной набор:

(61 ; 3), (8 ; 8), (24 ; 2), (35 ; 1), (66 ; 10), (88 ; 6), (2 ; 8), (72 ; 14), (21 ; 11), (64 ; 2), (78 ; 10), (3 ; 3), (86 ; 4), (78 ; 14)

Шаг 1: 61

Шаг 2: 8#61

Шаг 3: 8#61##24

Шаг 4: 8#61##24#####35

Шаг 5: 668##61####24#####35

Шаг 6: 66888#61####24#####35

Шаг 7: 66888261####24#####35

Шаг 8: 7266888###261#####24#####35

Шаг 9: 722188866##2#61#####24#####35

Шаг 10: 722188866##2#61#####2464#####35

Шаг 11: 722178866#882#61#####2464#####35

Шаг 12: 722178866#882#61#####3##2464#####35

Шаг 13: 722178866#882#61###86##3##2464#####35

Шаг 14: 72217886678882#61###86##3##2464#####35

Для полученного дерева исключить элемент 88

Шаг 1: 72217886678862#61####88#3##2464#####35

Шаг 2: 72217886678862#61#####3##2464#####35

### Графики оценки сложности.

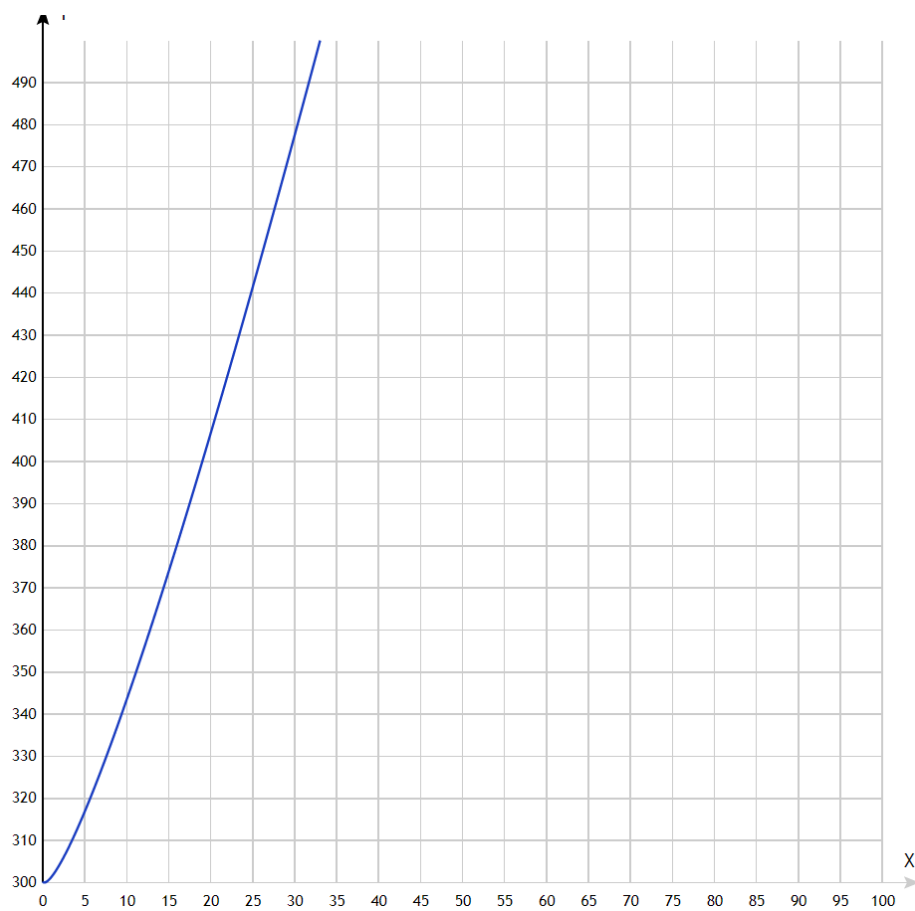


Рисунок 1 - первый график



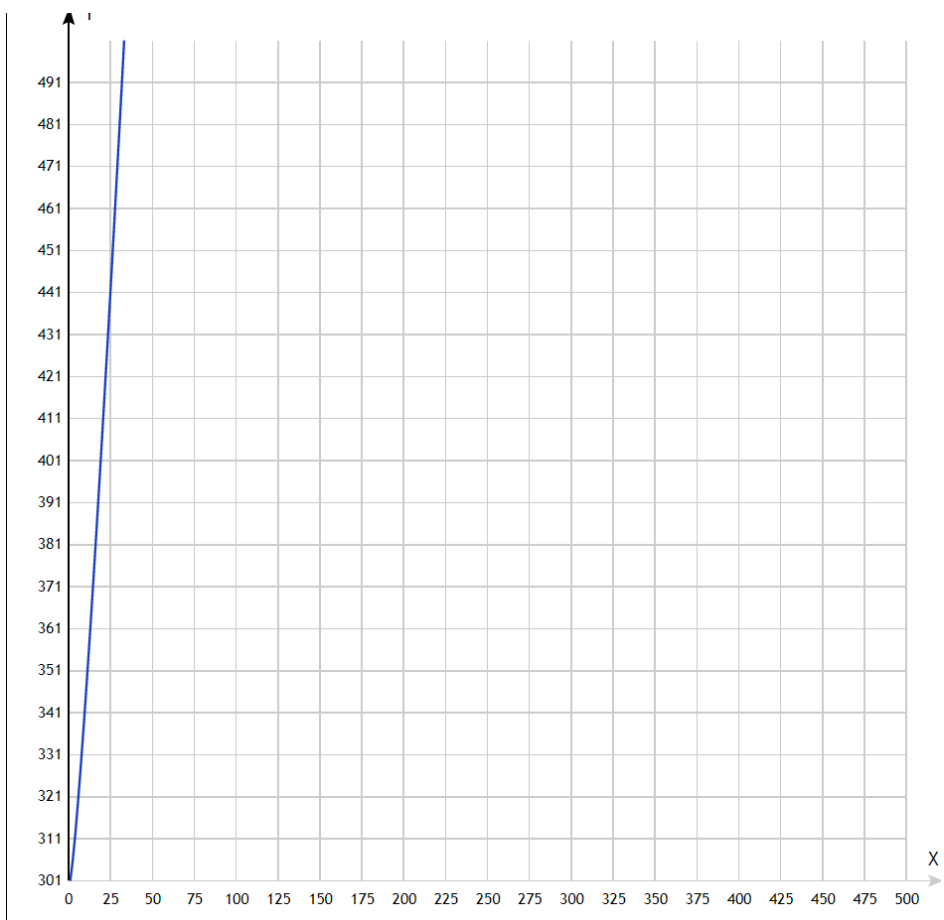


Рисунок 2 - второй график