BLG335E

ANALYSİS OF ALGORİTHMS I

CRN:10824

PROJECT 02

INSTRUCTOR:

ZEHRA ÇATALTEPE

STUDENT:

HÜSEYİN ERDOĞAN

040100054

Introduction

Aim of this homework is learning how to build priority queue using the heap data structure.. First program reads the “bids.txt” file and according to value of m(operation number) and p (probability) program does insert and update operations. After termination, program creates an output file named “output.txt”. This file consists information that which bid is maximum after every 100 operation. New bid and updated bid numbers are also written in this file.

Development and Runtime environment

Project is developed in Microsoft Windows 7 . Microsoft Visual, Dev-C++ and g++ are used for compiling.

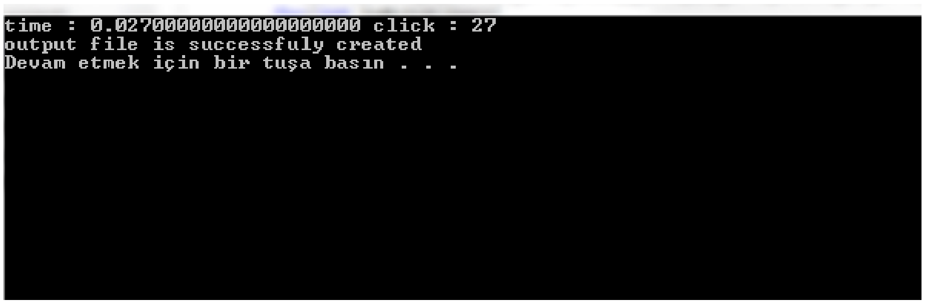
Program has one class. This class consists priority queue and heap sort functions.

In priority_queue class:

- `int *pq_array` **: for storing bids in array**
- `int heap_size` **: for storing heap size**
- `priority_queue(int)` **: for defining size of array and assigning heap size to “0”**
- `int parent(int)` **: for finding parent**
- `int left_child(int)`**: for finding left child**
- `int right_child(int)`**: for finding right child**
- `int length()`**: for returning size of array**
- `void max_heapify(int)` **: for organizing heap tree**

- `void build_max_heap()`: **for building heap tree**
- `void heapsort()` **: for doing heap sort**
- `int heap_maximum()` **: for returning maximum bid without deleting from tree**
- `int heap_extrac_max()`: **for returning maximum bid and deleting from tree**
- `void heap_increase_key(int,int)`: **for increasing bid that is in a specific location**
- `void max_heap_insert(int)`: **for inserting bid to heap**

When program is started, user enters to command line m(operation number) and p (probability). Program takes "bids.txt" file and after execution creates "output.txt" file. When program is terminated time is showed on the top and give information about output file.



```
time : 0.0270000000000000000 click : 27
output file is successfuly created
Devam etmek için bir tuşa basın . . .
```

In this lines I take m and p from command line .

```
int main(int argc, char* argv[]){

    int m = atoi(argv[1]);
    double p = atoi(argv[2]);
```

User have to write like this format to command line :

```
./studentID_AoA1_P2 m p
```

```
./040100054_AoA1_P2 10000 0.4
```

Ruquired Questions in Report

1) In this project we use max_heapify, max_heap_insert, heap_increase_key and heap_extrac_max functions.

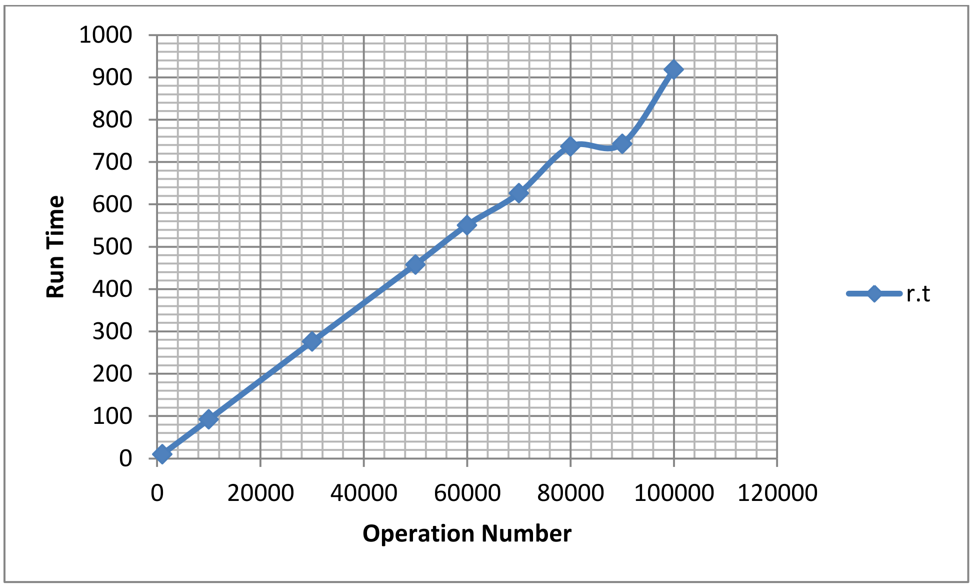
Running time of :

- max_heapify : T(n) = T(2n/3) + $\Theta(1)$ case 2 of master method

  T(n) = O(lg n)
- max_heap_insert : T(n) = O(2) + O(lg n)
- heap_increase_key : T(n) = O(3) + O(4*lg n)
- heap_extrac_max : T(n) = O(6) + O(lg n)

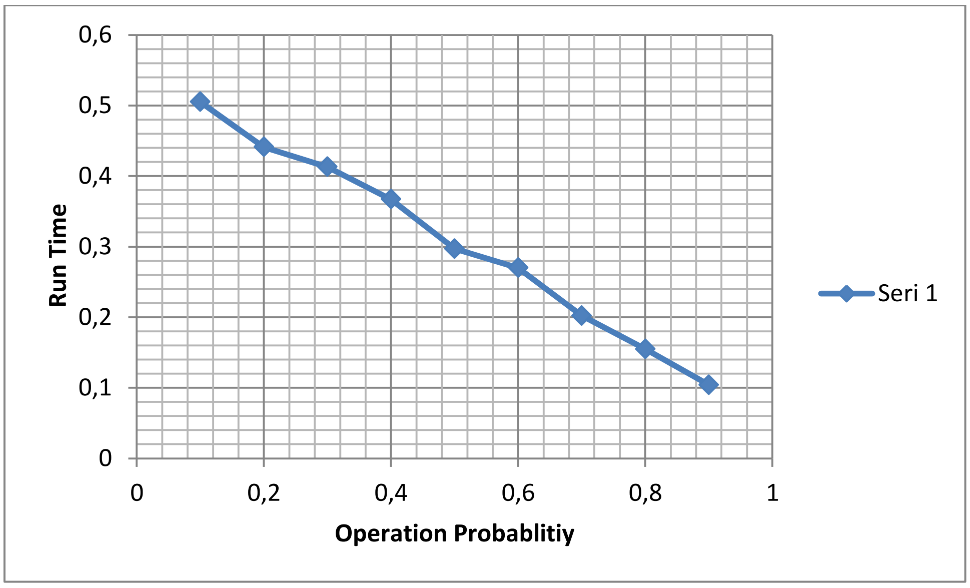
2) p=0.2

Operation Num	1000	10000	30000	50000	60000	70000	80000	90000	100000
Run Time(milisecond)	1	92	276	458	551	626	737	743	918



4) m=50000

Operation Probality	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Run Time(second)	0.505	0.441	0.413	0.367	0.297	0.270	0.202	0.155	0.104



5) Graph is directly efected by p (probability). Because according to p, program take a new bid or update a random bid. For new bid we use `void max_heap_insert(int)` function and increase heap size. For updating random bid we use `void heap_increase_key(int,int)` function and heap size is not changed. Because changing on heap size and differend functions, p effects runtime.