**Министерство образования и науки Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики   
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

**ОТЧЁТ**

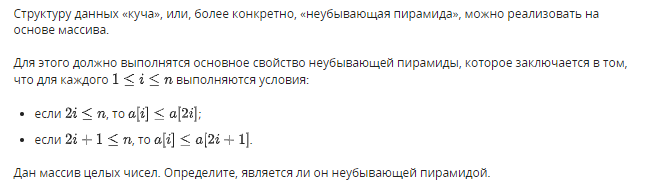
по лабораторной работе №5

Студент Кочарян С.А. группы P3218

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2019

## Куча ли?



#include "edx-io.hpp"

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

int main()

{

int quantity;

io >> quantity;

long \*numbers = new long[quantity];

for (int i = 0; i < quantity; i++) {

io >> numbers[i];

}

bool answer = true;

for (int i = quantity / 2 - 1; i >= 0 && answer; i--) {

if ((i + 1) \* 2 - 1 < quantity && numbers[(i + 1) \* 2 - 1] < numbers[i])

answer = false;

if ((i + 1) \* 2 < quantity && numbers[(i + 1) \* 2] < numbers[i])

answer = false;

}

if (answer)

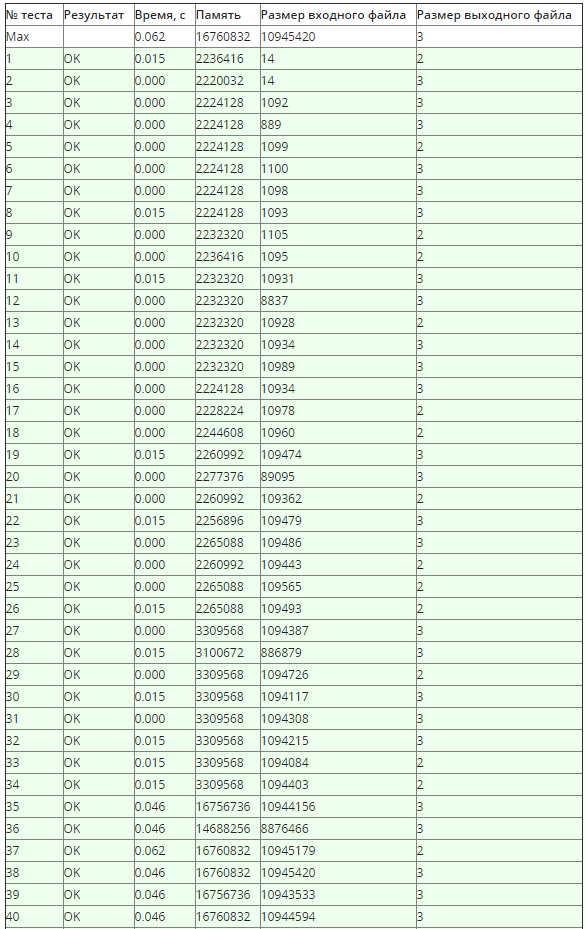
io << "YES";

else

io << "NO";

return 0;

}



## Очередь с приоритетами



#include "edx-io.hpp"

using namespace std;

struct number {

long value;

long order;

};

void swap(number\* a, number\* b) {

number temp = \*a;

\*a = \*b;

\*b = temp;

}

void heapUp(number\* array, long i, long\* positions) {

long parent = i / 2 - (1 - i % 2);

while (i != 0 && array[i].value < array[parent].value) {

positions[array[i].order] = parent;

positions[array[parent].order] = i;

swap(&array[i], &array[parent]);

i = parent;

parent = i / 2 - (1 - i % 2);

}

}

void heapDown(number\* array, long end, long\* positions) {

long position = 0;

long left = 2 \* (position + 1) - 1;

long right = 2 \* (position + 1);

while (position != end && ((array[position].value > array[left].value && left <= end) || (array[position].value > array[right].value && right <= end))) {

if (array[right].value < array[left].value && right <= end) {

positions[array[position].order] = right;

positions[array[right].order] = position;

swap(&array[right], &array[position]);

position = right;

}

else {

positions[array[position].order] = left;

positions[array[left].order] = position;

swap(&array[left], &array[position]);

position = left;

}

left = 2 \* (position + 1) - 1;

right = 2 \* (position + 1);

}

}

int main() {

long N;

io >> N;

number\* Queue = new number[N];

long head = 0;

long end = -1;

///////////////////

long\* array = new long[N];

char command;

long a, temp;

for (long i = 0; i < N; i++) {

io >> command;

switch (command)

{

case 'A':

io >> a;

Queue[++end] = number{ a, i };

array[i] = end;

heapUp(Queue, end, array);

break;

case 'X':

if (head > end) {

io << '\*' << '\n';

}

else {

array[Queue[head].order] = end;

array[Queue[end].order] = head;

swap(&Queue[head], &Queue[end]);

io << Queue[end--].value << '\n';

heapDown(Queue, end, array);

}

break;

case 'D':

io >> a;

io >> temp;

Queue[array[a - 1]].value = temp;

heapUp(Queue, array[a - 1], array);

break;

default:

break;

}

}

return 0;

}

