

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №4

З предмету «Алгоритми та структури даних»

Виконав Студент групи КН-36а

Рубан Ю.Д.

Перевірила:

ас. Бородіна І. О.

Харків 2017

АЛГОРИТМИ СОРТУВАННЯ

Мета роботи: познайомитися алгоритмами сортування та бінарним пошуком.

Завдання

Розробити програму, яка читає з клавіатури числа N , M ($1 < N, M < 256$); послідовність N ключів (цілих або дійсних чисел в залежності від варіанту завдання); послідовність M ключів. Програма зберігає першу послідовність до масиву та виконує сортування. Потім програма виводить відсортовану послідовність на екран та виконує бінарний пошук кожного елементу другої послідовності x : для кожного x повідомити, чи є він у першій послідовності, а якщо є, то на якому місці.

Варіант завдання

Сортування купою

Хід виконання роботи

Розроблено програму, яка читає з клавіатури числа N , M ($1 < N, M < 256$); послідовність N ключів цілих числа; послідовність M ключів. Програма зберігає першу послідовність до масиву та виконує сортування. Потім програма виводить відсортовану послідовність на екран та виконує бінарний пошук кожного елементу другої послідовності x : для кожного x повідомити, чи є він у першій послідовності, а якщо є, то на якому місці.

Код програми:

```
#include<iostream>
using namespace std;
int *fixHeap(int *a, int root,int bottom);
int* heapSort(int *a, int size);
void swap(int &a, int&b);
bool binSearch(int *A, int size, int x);
int main()
{
    int N, M;
    cout << "enter n, m" << endl;
    cin >> N >> M;
    int *a = new int[N];
    for (int i = 0; i < N; i++)
    {
        cin >> a[i];
    }
    cout << "initial array: " << endl;
    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
    a = heapSort(a, N);
    cout << "sorted array: " << endl;
    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
    int k;
    for (int i = 0; i < M; i++)
    {
        cout << "enter key for search" << endl;
        cin >> k;
        if (binSearch(a, N, k))
        {
            cout << "this elem exist in array" << endl;
        }
        else
        {
            cout << "no elem" << endl;
        }
    }
    cout << endl;

    system("pause");
    return 0;
}
void swap(int &a, int&b)
{
    int temp = a;
    a = b;
    b = temp;
}
int *fixHeap(int *a, int root,int bottom)
{
    int maxChild;
    bool done=false;
    while (2 * root+1 <= bottom && !(done))
    {
```

```

        if (2 * root+1 == bottom)
        {
            maxChild = 2*root+1;
        }
        else if (a[2 * root+1] > a[2 * root + 2])
        {
            maxChild = 2 * root+1;
        }
        else
        {
            maxChild = 2 * root+2;
        }
        if (a[root] < a[maxChild])
        {
            swap(a[root], a[maxChild]);
            root = maxChild;
        }
        else { done = true; }
    }

    return a;
}

void show(int*a, int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
}

int *heapSort(int *a, int size)
{
    for (int i = (size / 2) - 1; i >= 0; i--)
    {
        a=fixHeap(a, i, size);
    }
    for (int i = size-1; i >= 1; i--)
    {
        swap(a[0], a[i]);
        a = fixHeap(a, 0,i-1);
    }
    return a;
}

bool binSearch(int *A, int size, int x)
{
    int mid;
    int first = 0;
    int last = size - 1;
    if (A[0] > x || A[last] < x) { return false; }
    while (first <= last)
    {
        mid = first + (last - first) / 2;
        if (A[mid] > x)
        {
            last = mid - 1;
        }
        else if (A[mid]<x)
        {
            first = mid + 1;
        }
        if (x == A[mid])
            return true;
    }
    return false;
}

```

Результат виконання:

```
enter n, m
15 5
4 7 3 9 10 14 1 2 6 8 6 5 4 3 12
initial array:
4 7 3 9 10 14 1 2 6 8 6 5 4 3 12
sorted array:
1 2 3 3 4 4 5 6 6 7 8 9 10 12 14
enter key for search
1
this elem exist in array
enter key for search
4
this elem exist in array
enter key for search
11
no elem
enter key for search
16
no elem
enter key for search
6
this elem exist in array
Для продовження натисніть будь-яку клавішу . . .
```

Висновки

У даній лабораторній роботі я познайомився з алгоритмом сортування купкою. Цей алгоритм працює за $O(n \log(n))$ і це означає, що це один із найшвидших алгоритмів сортування. Також я познайомився з бінарним пошуком. За допомогою цього алгоритму можливо знаходити елементи масиву швидше ніж за допомогою лінійного пошуку.