повний архів

Сортування елементів масиву

(методи сортування, сортування перестановкою, вибором, швидке сортування, задача кількість різних чисел в масиві)

Розглянемо способи сортування. Сама тема сортування є однією з найбільш досліджених задач.

Є три способи сортування масивів:

$1- сортування вибором;

$1- сортування обміном;

$1- сортування вставкою.

Для кожного способу є багато алгоритмів, які відрізняються часом сортування, який злежить від числа операцій порівняння і операцій обміну.

Традиційно розрізняють внутрішнє сортування, яке обробляє дані оперативної пам’яті, і зовнішнє сортування, яке оперує з даними розміщеними на дисках.

Розглянемо сортування числового одномірного масиву.

Відсортувати числовий масив: 7, 3, 8, 4,8, 5, 9, 1.

Звичайне сортування : 1, 3, 4, 5, 7, 8, 8, 9,.

Адресне сортування: 7, 3, 8, 4, 8, 5, 9, 1.

5, 2, 6, 3, 7, 4, 8, 1 (адреса).

Є багато різноманітних алгоритмів сортування (сортування бульбашкою, сортування за допомогою дерева, пірамідальне сортування, швидке сортування (половинного поділу).

Розглянемо деякі з них в дещо видозміненому вигляді.

Сортування вектором

```
#include "fstream"
 #include<algorithm>
 #include<vector>


using namespace std;
 int main()
 {ifstream cin("E.dat");
 ofstream cout("E.sol");
 long long n,k,l,i,j,c;
 cin>>n>>k;
 vector<long long int> a(n);
```

```
 for(i=0;i<n;i++) cin>>a[i];
 sort(a.begin(),a.end());





l=0;
 for(i=k;i<n;i++)
 {l=l+a[i];



}
 cout<<l;
 return 0;
 }
```

Перебір вектором

```
 #include <iostream>
#include <fstream>
#include <math.h>
#include <vector>
#include <algorithm>

using namespace std;
vector <int> a;

 ifstream f;
 ofstream g;

void printper(int n);
int main()
{
    f.open("input.dat");
    g.open("output.ans");
    int n;
    f >> n;
```

```
    for (int i=1;i<=n;i++){
        a.push_back(i);
    }
    printper(n);

    while (next_permutation(a.begin(),a.end())){
        printper(n);
    };
    //printper(n);

    f.close();
    g.close();
    return 0;
}

void printper(int n)
{
    for (int i=0;i<n-1;i++){
        g << a[i] << " ";
    }
    g << a[n-1] << endl;
}
```