

BUỔI THỰC HÀNH 4: THUẬT TOÁN COUNTING SORT

4.1. Bài thực hành 1: Quản lý danh sách đặc 100 phân tử kiểu số nguyên (int)

4.1.1. Khai báo cấu trúc danh sách.

4.1.2. Viết thủ tục nhập danh sách.

4.1.3. Viết thủ tục xuất danh sách

4.1.4. Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán Counting Sort

Hướng dẫn:

```
#include<iostream>

using namespace std;

void nhapmang(int a[], int &n, int &k)
{
    do{
        cout<<"\n So luong phan tu mang:";
        cin>>n;
    }while(n <=0 || n > 100);

    for(int i = 0; i < n; i++)
    {
        cout<<"\na["<<i<<"]="";
        cin>>a[i];
    }
    for(int i = 0; i < n; i++)
    {
```

```

        if(k < a[i])
            k = a[i];
    }
}

void countingsort(int a[], int b[], int k, int n, int & kt)
{
    int i, j;
    int c[100];
    for(i = 0; i <= k; i++)
    {
        c[i] = 0;
    }
    for(j = 0; j < n; j++)
    {
        c[a[j]]++;
    }
    kt = 0;
    for(j = 0; j <= k; j++)
    {
        while(c[j]>0)
        {
            b[kt++] = j;
            c[j]--;
        }
    }
}

void xuatmang(int b[], int kt)

```

```

    {
        cout<<"Mang a sau khi sap tang:\n";
        for(int i = 0; i <kt; i ++)
            cout<<b[i];
    }
void main()
{
    int n, k=0,kt;
    int a[100];
    int b[100];
    nhapmang(a, n,k);
    countingsort(a, b, k, n,kt);
    xuatmang(b, kt);
    system("pause");
}

```

4.2. Bài thực hành 2: Quản lý danh sách đặc 100 phân tử kiểu số nguyên (int)

4.2.1. Khai báo cấu trúc danh sách.

4.2.2. Viết thủ tục nhập danh sách.

4.2.3. Viết thủ tục xuất danh sách

4.2.4. Viết thủ tục sắp xếp danh sách theo thứ tự giảm dần bằng thuật toán

Counting Sort

Hướng dẫn: *Sinh viên tự cài đặt*