

BUỔI THỰC HÀNH 3: THUẬT TOÁN QUICK SORT

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

3.1.1. Khai báo cấu trúc danh sách.

3.1.2. Viết thủ tục nhập danh sách.

3.1.3. Viết thủ tục xuất danh sách

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

Hướng dẫn:

```
void nhapmang(int a[], int &n)
{
    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];
    }
}

void hoanvi(int &x, int &y)
{
```

```
    int tam = x;
    x = y;
    y = tam;
}
```

```
void QuickSort(int a[], int left, int right)
{
    int    i, j, x;
    x = a[(left+right)/2]; //Chọn phần tử giữa làm giá trị mốc
    i = left;
    j = right;
    while(i <= j)
    {
        while(a[i] < x) i++;
        while(a[j] > x) j--;
        if(i <= j) {
            hoavi(a[i], a[j]); //đổi chỗ a[i] và a[j]
            i++; j--;
        }
    }
    if(left < j)
        QuickSort(a, left, j);
    if(i < right)
        QuickSort(a, i, right);
}
```

```

void xuatmang(int a[], int n)
{
    cout<<"Mang a sau khi sap tang:\n";
    for(int i = 0; i < n; i ++)
        cout<<a[i];
}

void main()
{
    int a[100], n,left,right;

    nhapmang(a, n);
    left = 0;
    right = n -1;
    QuickSort(a, left,right);
    xuatmang(a, n);
    system("pause");
}

```

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

3.2.1. Khai báo cấu trúc danh sách.

3.2.2. Viết thủ tục nhập danh sách.

3.2.3. Viết thủ tục xuất danh sách

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

Hướng dẫn: *Sinh viên tự cài đặt*