Bài thực hành

Bài thực hành 4: Cài đặt QuickSort

Quick sort là thuật toán sắp xếp, hoạt động theo cách sau: Chọn một phần tử trong mảng làm điểm đánh dấu và sau đó chia mảng thành hai mảng con bằng cách so sánh các phần tử trong mảng với điểm đánh dấu. Mảng 1 sẽ chứ các phần tử nhỏ hơn hoặc bằng điểm đánh dấu và mảng 2 sẽ gồm các phần tử lớn hơn điểm đánh dấu.

**Cài đặt**

* Chọn điểm đánh dấu cho mảng, chọn điểm đánh dấu là số cuối cùng của mảng.
* Tạo hai biến là trái và phải để trỏ tới bên trái và bên phải của danh sách.
* Thực hiện so sánh các phần tử với điểm đánh dấu. Nếu phần tử nhỏ hơn điểm đánh dấu thì dịch chuyển qua bên trái và ngược lại.
* Sau khi dịch chuyển thực hiện công việc sắp xếp các phần tử trong mảng con mới, trước khi tiếp tục phân đoạn tiếp theo.

**Code**

#include<iostream>

using namespace std;

void swap(int arr[] , int pos1, int pos2){

int temp;

temp = arr[pos1];

arr[pos1] = arr[pos2];

arr[pos2] = temp;

}

int partition(int arr[], int low, int high){

int p = arr[high];

int left = low;

int right = high - 1;

while(true) {

while(left <= right && arr[left] < p) left++; //tìm phần tử >= phần tử p trong mảng

while(right >= left && arr[right] > p) right--; //tìm phần tử <= phần tử p trong mảng

if(left >= right) break; //sau khi duyệt xong thì thoát khỏi vòng lặp

swap(arr[left], arr[right]); //nếu chưa thì sử dụng hàm swap để tráo đổi

left++;

right--;

}

swap(arr[left], arr[high]);

return left; //trả về chỉ số sẽ dùng để chia đôi mảng

}

void quickSort(int arr[], int low, int high){

if(low < high){

int pos = partition(arr, low, high);

//gọi đệ quy sắp xếp 2 mảng con trái và phải

quickSort(arr, low, pos-1);

quickSort(arr, pos+1, high);

}

}

int main()

{

int n ;

cout << "Nhap so phan tu cua mang: ";

cin>>n;

int arr[n];

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

cin >> arr[i];

}

quickSort(arr, 0 , n-1);

cout<<"Mang duoc sap xep la: ";

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

cout<< arr[i]<<"\t";

}

return 0;

}

**Kết quả**

A black screen with white text

Description automatically generated with low confidence