**Bài toán cài đặt quicksort**

Đây là đoạn code thực hiện thuật toán QuickSort để sắp xếp một mảng số nguyên. Các bước thực hiện như sau:

- Hàm `swap` để hoán đổi giá trị của hai phần tử.

- Hàm `partition` để chọn giá trị pivot và phân chia mảng thành hai phần: phần bên trái pivot có giá trị nhỏ hơn pivot và phần bên phải pivot có giá trị lớn hơn pivot.

- Hàm `quickS` để triển khai thuật toán QuickSort. Ban đầu, hàm nhận mảng a và chỉ số l và r (là chỉ số đầu và cuối của mảng cần sắp xếp). Nếu chỉ số đầu lớn hơn chỉ số cuối, hàm kết thúc. Nếu không, hàm thực hiện chia phần tử pivot và sắp xếp đệ quy mảng bên trái và bên phải của pivot.

Input: Một mảng số nguyên a chưa được sắp xếp và chỉ số l, r của mảng.

Output: Mảng a đã được sắp xếp tăng dần.

#include<stdio.h>

void swap(int a[1000], int x, int y){

int temp = a[x];

a[x] = a[y];

a[y] = temp;

}

int partition(int a[1000], int l, int r){

// Gia tri pỉvot bang gia tri phan tu dau tien

int pivot = a[l];

int count = 0;

// Dem so luong phan tu nam ben trai pivot

for(int i=l+1;i<=r;i++){

if(a[i]<=pivot)

count++;

}

// Lay vi tri cho gia tri pivot

int pvI=l+count;

//Hoan vi

swap(a,l,pvI);

int i=l;

int j=r;

// Doi cho phan tu ben trai lon hon pivot va phan tu ben phai be hon pivot

while(i<pvI && j>pvI){

while(a[i]<=pivot){

i++;

}

while(a[j]>pivot){

j--;

}

if(i<pvI && j>pvI){

swap(a,i,j);

}

}

return pvI;

}

void quickS(int a[1000],int l, int r){

if(left > right)

{

return;

}

// Chia

int pivot = partition(a,l,r);

// Tri gom

// Sap xep day ben trai

quickS(a,l,pivot-1);

// Sap xep day ben phai

quickS(a,pivot+1,r);

}