**Trần Hoàng Triển - MSSV:46.01.104.196**

**Cài đặt bài toán Quicksort**

**Quick Sort**

**Dữ liệu vào:**

Nhập vào n là dãy số nguyên chưa sắp xếp

A picture containing text, font, white, screenshot

Description automatically generated

**Dữ liêu ra :**

* Dãy số đã được sắp xếp

A screenshot of a phone

Description automatically generated with low confidence

**Miêu tả cách xử lý dữ liệu như sau:**

Khi nhập dữ liệu là "27 9 20 27", chương trình sẽ thực hiện các bước sau:

1. Sau khi n nhập số lượng phần tử (`n`).

2. Tạo một mảng có kích thước `n` để lưu trữ các phần tử.

3. Sử dụng vòng lặp `for` để lấy giá trị từng phần tử và lưu vào mảng.

- Lần lượt, bạn nhập giá trị "27", "9", "10", "27".

- Các giá trị này được lưu vào mảng theo thứ tự tương ứng.

4. Sau khi nhập xong, chương trình gọi hàm `quickSort(arr, 0, n - 1)` để sắp xếp mảng.

5. Trong hàm `quickSort`, mảng `arr` được chia thành các phần tử nhỏ hơn và lớn hơn một phần tử chốt (pivot).

6. Tiếp theo, các mảng con được sắp xếp đệ quy bằng cách gọi lại hàm `quickSort` trên mảng con trái và mảng con phải của pivot.

7. Quá trình chia và sắp xếp đệ quy diễn ra cho đến khi các mảng con chỉ chứa một phần tử hoặc không có phần tử nào.

8. Sau khi quá trình sắp xếp kết thúc, chương trình gọi hàm `printArray(arr, n)` để in ra mảng đã sắp xếp.

- Kết quả in ra là "9 10 22 27".

**Để thực thi chương trình trên, bạn có thể làm theo các bước sau:**

1. Mở trình biên dịch C++ (như Dev-C++, Visual Studio,).

2. Tạo một tệp tin mới và sao chép đoạn mã vào tệp tin đó.

3. Lưu tệp tin với phần mở rộng `.cpp` (ví dụ: `**Quicksort.cpp**`).

4. Tiến hành chạy chương trình và nhập dữ liệu vào n là dãy số nguyên chưa sắp xếp.

5. Trình biên dịch sẽ thực thi chương trình và hiển thị kết quả.

**Source code:**

#include<stdio.h>

#include<iostream>

using namespace std;

//tạo hàm swap để tráo đổi các vị trí

void swap(int &a, int &b)

{

int t = a;

a = b;

b = t;

}

// phân đoạn trong mảng

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

{

int pivot = arr[high]; // khai báo phần tử đánh dâu pivot

int left = low; //khai báo biến bên trái

int right = high - 1; //khai báo biến bên phải

while(true){

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

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

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

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

left++; // Vì left hiện tại đã xét, nên cần tăng

right--; // Vì right hiện tại đã xét, nên cần giảm

}

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

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

}

/\* Hàm thực hiện giải thuật quick sort \*/

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

{

if (low < high)

{

// index là chỉ số nơi phần tử này đã đứng đúng vị trí và đây là phần tử chia mảng làm 2 mảng con trái & phải

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

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

quickSort(arr, low, index - 1);

quickSort(arr, index + 1, high);

}

}

/\* Hàm xuất mảng \*/

void printArray(int arr[], int size)

{

int i;

for (i=0; i < size; i++){

cout << arr[i];

cout<<"\t";

}

}

int main()

{

int n;

cin >> n;

int arr[n];

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

cin >> arr[i];

quickSort(arr, 0, n-1);

cout<<"Mảng sau khi được sắp xếp: \n";

printArray(arr,n);

}