**Trần Hoàng Triển - MSSV:46.01.104.196**

**Cài đặt bài toán Quicksort với vị trí mốc ngẫu nhiên**

**Dữ liệu vào:**

Nhập vào n là dãy số nguyên chưa sắp xếp

A picture containing text, font, screenshot, white

Description automatically generated

**Dữ liêu ra :**

Dãy số đã được sắp xếp với mốc ngẫu nhiên

A picture containing text, font, white, screenshot

Description automatically generated

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

Khi nhập dữ liệu "n = 9, -3, 5, 2, 6, 8, -6, 1, 3" vào chương trình, quá trình xử lý dữ liệu được thực hiện như sau:

1. Sau khi nhập giá trị của n là 9,

2. Sau đó, bạn nhập các giá trị của mảng theo thứ tự: -3, 5, 2, 6, 8, -6, 1, 3. Chương trình lưu các giá trị này vào mảng arr.

3. Tiếp theo, chương trình sử dụng hàm srand(time(NULL)) để khởi tạo seed cho hàm rand(), đảm bảo rằng mỗi lần chạy chương trình, các số ngẫu nhiên được tạo ra bởi hàm rand() là khác nhau.

4. Hàm quickSort(arr, 0, n - 1) được gọi để sắp xếp mảng arr theo thứ tự tăng dần. Tham số đầu tiên của hàm quickSort là mảng arr, tham số thứ hai là chỉ số bắt đầu (0) và tham số thứ ba là chỉ số kết thúc (n - 1).

5. Trong hàm quickSort, một vị trí mốc ngẫu nhiên được chọn bằng cách gọi hàm getRandomPivot(arr, low, high). Trong trường hợp này, vị trí mốc ngẫu nhiên sẽ là một số ngẫu nhiên trong khoảng từ low đến high (tức là từ 0 đến n - 1).

6. Sau đó, quá trình phân đoạn được thực hiện bằng cách chọn phần tử mốc và xác định các phần tử nhỏ hơn mốc ở bên trái và các phần tử lớn hơn mốc ở bên phải.

7. Tiếp theo, hai đệ quy quickSort được gọi để sắp xếp các mảng con trái và phải của phần tử mốc. Quá trình này được thực hiện đệ quy cho đến khi các mảng con chỉ còn một phần tử.

8. Cuối cùng, mảng sau khi được sắp xếp được in ra màn hình bằng cách gọi hàm printArray(arr, n).

Với dữ liệu "n = 9, -3, 5, 2, 6, 8, -6, 1, 3", kết quả in ra sẽ là: "-6, -3, 1, 2, 3, 5, 6, 8, 9", tương ứng với mảng đã được sắp xếp theo thứ tự tăng

**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: `**QuicksortRandom.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 <iostream>

#include <cstdlib> // thư viện để sử dụng hàm rand()

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;

}

// Chọn một vị trí mốc ngẫu nhiên

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

{

int randomIndex = low + rand() % (high - low + 1);

return randomIndex;

}

// Phân đoạn trong mảng

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

{

int pivotIndex = getRandomPivot(arr, low, high);

int pivot = arr[pivotIndex];

// Đưa phần tử pivot về cuối mảng

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

int left = low;

int right = high - 1;

while (true) {

while (left <= right && arr[left] < pivot)

left++;

while (right >= left && arr[right] > pivot)

right--;

if (left >= right)

break;

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

left++;

right--;

}

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

return left;

}

// Hàm thực hiện giải thuật QuickSort

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

{

if (low < high) {

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

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

quickSort(arr, low, pivotIndex - 1);

quickSort(arr, pivotIndex + 1, high);

}

}

// Hàm xuất mảng

void printArray(int arr[], int size)

{

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

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

}

cout << endl;

}

int main()

{

int n;

cin >> n;

int arr[n];

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

cin >> arr[i];

srand(time(NULL)); // Khởi tạo seed cho hàm rand() dựa trên thời gian hiện tại

quickSort(arr, 0, n - 1);

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

printArray(arr, n);

return 0;

}