**Bài thực hành**

**Bài thực hành 5: Cài đặt QuickSort với vị trí mốc ngẫu nhiên**

Trong giải thuật QuickSort, ta chọn một phần tử làm "mốc" (pivot) và sắp xếp các phần tử trong mảng thành hai phần: một phần chứa các phần tử có giá trị nhỏ hơn pivot và một phần chứa các phần tử có giá trị lớn hơn pivot. Sau đó, ta tiếp tục sắp xếp đệ quy các phần đó.

Input: Nhập vào mảng

Output: Xuất mảng đã sắp xếp

**Code**

#include <iostream>

#include <cstdlib>

using namespace std;

void swap(int& a, int& b) {

int temp = a;

a = b;

b = temp;

}

int partition(int arr[], int left, int right, int pivot) {

int pivotValue = arr[pivot];

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

int Index = left;

for (int i = left; i < right; i++) {

if (arr[i] < pivotValue) {

swap(arr[i], arr[Index]);

Index++;

}

}

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

return Index;

}

int randPivot(int left, int right) {

return left + rand() % (right - left + 1);

}

void quickSort(int arr[], int left, int right) {

if (left >= right) {

return;

}

int pivot = randPivot(left, right);

pivot = partition(arr, left, right, pivot);

quickSort(arr, left, pivot - 1);

quickSort(arr, pivot + 1, right);

}

void printArray(int arr[], int size) {

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

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

}

cout << endl;

}

int main() {

int arr[] = { 5, 2, 6, 1, 3, 9, 4 };

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Mảng đã nhập:" << endl;

printArray(arr, size);

quickSort(arr, 0, size - 1);

cout << "mảng đã sắp xếp:" << endl;

printArray(arr, size);

return 0;

}

**Kết quả**

**A screen shot of a computer

Description automatically generated with medium confidence**