

🔍 Search

Write a quicksort program with cpp.

- 🌀 ChatGPT
- 🔗 Explore GPTs

- Projects
- 📁 New project

- Chats
- C++ Quicksort Implementati...

Here's a simple and clean implementation of the **Quicksort** algorithm in **C++**:

cpp Copy

```
#include <iostream>
#include <vector>

// Partition function
int partition(std::vector<int>& arr, int low, int high) {
    int pivot = arr[high]; // Pivot element
    int i = low - 1;       // Index of smaller element

    for (int j = low; j < high; ++j) {
        if (arr[j] < pivot) {
            ++i;
            std::swap(arr[i], arr[j]);
        }
    }
    std::swap(arr[i + 1], arr[high]); // Place pivot in correct position
    return i + 1;
}

// Quicksort function
void quicksort(std::vector<int>& arr, int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high); // Partition index

        quicksort(arr, low, pi - 1); // Sort left part
        quicksort(arr, pi + 1, high); // Sort right part
    }
}

// Helper to print the array
void printArray(const std::vector<int>& arr) {
    for (int num : arr)
        std::cout << num << " ";
    std::cout << "\n";
}

// Main function
int main() {
    std::vector<int> data = {10, 7, 8, 9, 1, 5};
    std::cout << "Unsorted array: ";
    printArray(data);

    quicksort(data, 0, data.size() - 1);

    std::cout << "Sorted array: ";
    printArray(data);
    return 0;
}
```

🔍 Key Points:

- **Time Complexity:** Average is $O(n \log n)$, worst is $O(n^2)$
- **Space Complexity:** $O(\log n)$ due to recursion stack
- **In-place:** Yes (doesn't require extra array)

Let me know if you want a version with randomized pivot or tail-recursion optimization!

📄 🔊 🍌 🗣️ ↺

Message ChatGPT