1. **代码实践**

**排序之冒泡、快排、二路归并：**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

/\*冒泡排序

void Sort(vector<int>& nums){

int n = nums.size();

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

//int cur = i, max = nums[i];

for(int j = 1; j < n - i; ++j){

if(nums[j] < nums[j - 1]){

int tmp = nums[j];

nums[j] = nums[j - 1];

nums[j - 1] = tmp;

}

}

}

}\*/

//快排

void Swap(vector<int>& v, int i, int j){

int tmp = v[i];

v[i] = v[j];

v[j] = tmp;

}

int Partition(vector<int>& v, int left, int right){ //分区

int pivot = left; //基准

int index = pivot + 1;

for(int i = index; i < right; ++i){ //左闭右开

if(v[i] < v[pivot]){

Swap(v, i, index);

++index;

}

}

Swap(v, pivot, index - 1); //index-1指向小于v[pivot]的最右侧的一个值

return index - 1; //返回中间值所在的位置

}

void QuickSort(vector<int>& v, int left = 0, int right = 0){

int part;

if(left < right){

part = Partition(v, left, right);

QuickSort(v, left, part); //左闭右开

QuickSort(v, part + 1, right);

}

}

/\*

// 归并排序(分治算法)

vector<int> Merge(vector<int> left, vector<int> right){

vector<int> result;

int i = 0, j = 0;

while(i < left.size() && j < right.size()){

if(left[i] <= right[j]){

result.push\_back(left[i]);

++i;

}

else{

result.push\_back(right[j]);

++j;

}

}

while(i < left.size()){

result.push\_back(left[i]);

++i;

}

while(j < right.size()){

result.push\_back(right[j]);

++j;

}

return result;

}

vector<int> Sort(vector<int> nums){

int n = nums.size();

if(n < 2) return nums;

int mid = floor(n / 2);

//cout << mid << endl;

vector<int>::iterator l = nums.begin() + mid;

vector<int> left(nums.begin(), l), right(l, nums.end());

return Merge(Sort(left), Sort(right));

}

\*/

int main(){

vector<int> nums;

int n;

cin >> n;

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

int tmp;

cin >> tmp;

nums.push\_back(tmp);

}

//vector<int> ans = Sort(nums);

//sort(nums.begin(), nums.end());

QuickSort(nums, 0, nums.size());

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

//cout << ans[i] << " ";

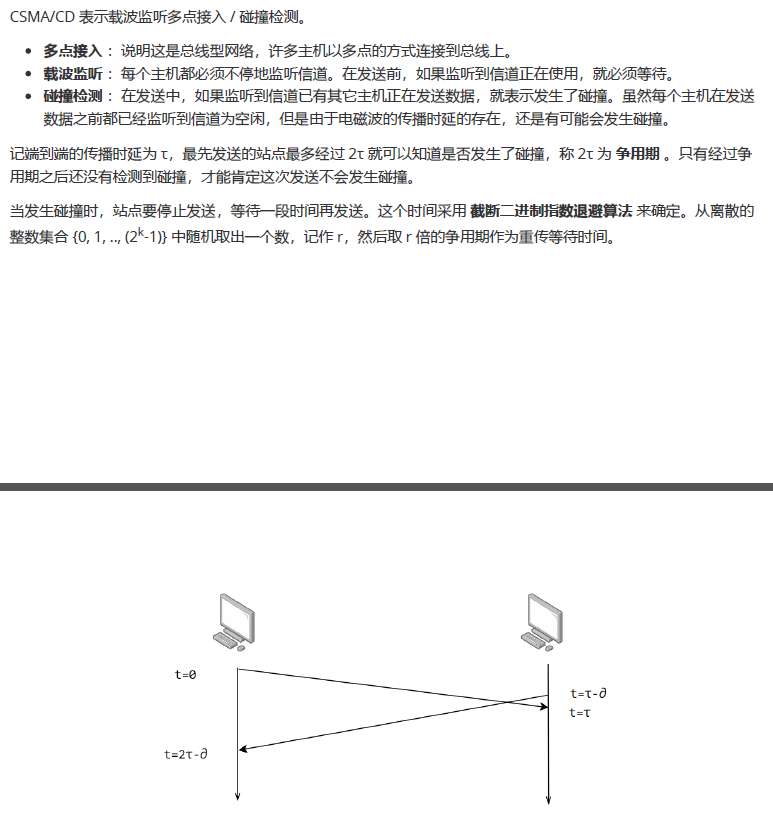
cout << nums[i] << " ";

}

return 0;

}

1. **计算机基础整理**



1. **开源特训营工作总结**
2. 将9月23日每日作业提交到Git仓库中。