

一、题目相关说明：

- 必须原地回形排序，包含两层意思：
 - ① 只能在接收输入数据的数组上进行操作，仅可使用除此之外的 $O(1)$ 辅助空间和 $O(\log n)$ 的递归栈空间；
 - ② 数组最后输出前，必须是回形序的数组，输出操作只能顺序输出；
 - 算法的最坏时间复杂度为 $O(n \log n)$ ，空间复杂度不超过 $O(\log n)$ ；
 - 可能的错误做法：
 - ① 正常升序排序数组，然后回形打印……………数组最后非回形序，判为0分；
 - ② 正常升序排序数组，然后借用辅助数组生成回形序，再输出……………空间复杂度为 $O(n)$ ，酌情扣分；
 - ③ 正常升序排序数组，然后用 $O(n^2)$ 的移动次数生成回形序……………最坏时间复杂度为 $O(n^2)$ ，可能过不了所有测试样例，酌情扣分；
 - ④ 没有改进pivot的选取方案……………最坏时间复杂度为 $O(n^2)$ ，可能过不了所有测试样例，酌情扣分；
-

二、框架代码：

- C:

```
#include <stdio.h>

void clip_sort(int* nums,int n) {
    // TODO
}

int main(){
    // input n
    int n = 0;
    scanf("%d",&n);
    // input nums
    int* nums = (int*)malloc(sizeof(int)*n);
    for(int i=0;i<n;i++){
        scanf("%d",nums+i);
    }

    // TODO: clip-sort
    clip_sort(nums,n);

    // output clip-sorted nums
    for(int i=0;i<n;i++){
        printf("%d",nums[i]);
        if(i<n-1)
            printf(" ");
    }
    // delete nums
```

```
free(nums);  
}
```

- C++:

- 不使用STL:

```
#include<iostream>  
using namespace std;  
  
void clip_sort(int* nums,int n) {  
    // TODO  
}  
  
int main(){  
    // input n  
    int n = 0;  
    cin >> n;  
    // input nums  
    int* nums = new int[n];  
    for(int i=0;i<n;i++){  
        cin >> nums[i];  
    }  
  
    // TODO: clip-sort  
    clip_sort(nums,n);  
  
    // output clip-sorted nums  
    for(int i=0;i<n;i++){  
        cout << nums[i];  
        if(i<n-1)  
            cout << ' ' ;  
    }  
    // delete nums  
    delete[] nums;  
}
```

- 使用STL:

```
#include<iostream>  
#include<vector>  
using namespace std;  
  
void clip_sort(vector<int>& nums) {  
    // TODO  
    int n = nums.size();  
}
```

```

int main(){
    // input n
    int n = 0;
    cin >> n;
    // input nums
    vector<int> nums(n,0);
    for(int i=0;i<n;i++){
        cin >> nums[i];
    }

    // TODO: clip-sort
    clip_sort(nums);

    // output clip-sorted nums
    for(int i=0;i<n;i++){
        cout << nums[i];
        if(i<n-1)
            cout << ' ';
    }
}

```

- **Java:**

```

import java.util.*;

public class ClipSort {
    static void clip_sort(int[] nums){
        // TODO
        int n = nums.length;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // input n
        int n = 0;
        n = sc.nextInt();
        // input nums
        int[] nums = new int[n];
        for(int i=0;i<n;i++){
            nums[i] = sc.nextInt();
        }

        // TODO: clip-sort
        clip_sort(nums);

        // output clip-sorted nums
        for(int i=0;i<n;i++){
            System.out.print(nums[i]);
            if(i<n-1)

```

```
        System.out.print(' ');  
    }  
}  
}
```

- Python:

```
def clip_sort(nums):  
    # TODO  
    n = len(nums)  
    return  
  
if __name__ == "__main__":  
    # input n  
    n = int(input())  
    # input nums  
    nums = [int(num) for num in input().split()]  
  
    # TODO: clip-sort  
    clip_sort(nums)  
  
    # output clip-sorted nums  
    for i in range(len(nums)):  
        print(nums[i],end='')  
        if i < n-1:  
            print(' ',end='')
```
