

作业4

●Count sort:

- 基本思想是对于列表 a 中的每个元素 $a[i]$ ，计算小于 $a[i]$ 的元素个数，将 $a[i]$ 插入到由 $count$ 决定的列表下标位置中，算法结束后，用临时列表覆盖原始列表

```
void Count_sort(int a[], int n) {
    int i, j, count;
    int* temp = malloc(n*sizeof(int));

    for (i = 0; i < n; i++) {
        count = 0;
        for (j = 0; j < n; j++)
            if (a[j] < a[i])
                count++;
            else if (a[j] == a[i] && j < i)
                count++;
        temp[count] = a[i];
    }

    memcpy(a, temp, n*sizeof(int));
    free(temp);
} /* Count_sort */
```

作业4

●Count sort:

➤ 问题:

- 如果我们试图并行化外层循环，哪些变量为 private，哪些变量为 shared？
- 是否存在循环携带的数据依赖性？为什么？
- 编写并行化的 Count_sort
- 并行化的 Count_sort 与串行化的 Count_sort 相比，性能如何？
- 并行化的 Count_sort 与串行化的 qsort 库函数相比，性能如何？