

# 改进冒泡排序伪代码

## 分析

优化二的 "无交换则提前停止" 实际上被优化三的 "设置排序边界为后续部分如果无交换则跳过排序" 完全包含，实际只需要添加三个优化。

## 伪代码

```
lower = 0, upper = length - 1
while (lower < upper){

    new_upper = lower
    max_value = arr[lower]
    for i from lower + 1 to upper - 1
        if (arr[i] > max_value)
            arr[i - 1] = max_value # 不动
            max_value = arr[i] # 新的最值
        else then
            arr[i - 1] = arr[i] # 将值前移
            new_upper = i # 如果出现交换，则认为前面的部分可能无序
    upper = new_upper

    new_lower = upper
    min_value = arr[upper]
    for i from upper - 2 to lower
        if (arr[i] < min_value)
            arr[i + 1] = min_value
            min_value = arr[i]
        else then
            arr[i + 1] = arr[i]
            new_lower = i
    lower = new_lower
}
```

## 有优化的快速排序

- 作业里面限制了实现方式，这种实现有一些细节
  - `last` 的边界问题。
  - `low` 和 `high` 随 `find_next` 和 `find_previous` 要更新
  - 由于区间开闭的设计问题，往下递归时需要使用 `next` 而不是 `prv`。
- 实际实现的几个点
  - `find_next` 和 `find_previous` 的逻辑直接在循环里面实现。
  - 将边界增长移动到外层循环开始，调整边界范围，以便退出循环时能够正确保留结果。

## 堆排序

先构造一个堆，不断把堆顶放到尾部，然后不断缩小堆的大小，直到堆为空，之后序列有序。

## 代码

```
int main(){
    int n;
    int a[20];
    cin >> n;
    for(int i=1; i<=n; i++){
        cin >> a[i];
    }
    auto access = [&a](int x){
        while(x!=1){
            if(a[x] > a[x >> 1]){
                swap(a[x], a[x >> 1]);
            }
            x >>= 1;
        }
    };
    auto put = [&](int x, int len){
        while((x << 1) <= len){
            int next=x;
            if(a[x << 1] > a[x]){
                swap(a[x << 1], a[x]);
                next = x << 1;
            }
            if( (x << 1) + 1 <= len && a[x] < a[(x << 1) + 1]){
                swap(a[(x << 1) + 1], a[x]);
                next = (x << 1) + 1;
            }
            if(next == x){
                break;
            }
            x = next;
        }
    };

    for(int i=n; i>=1; i--){
        access(i);
    }
    for(int i=n; i>=1; i--){
        swap(a[1], a[i]);
        put(1, i - 1);
    }
    for(int i=1; i<=n; i++){
        cout << a[i] << " ";
    }
    return 0;
}
```

## 运行

```
PS C:\Users\huany\Desktop\work_space\ds> ./mai
12
34 15 65 59 79 42 40 80 50 61 23 46
15 23 34 40 42 46 50 59 61 65 79 80
PS C:\Users\huany\Desktop\work_space\ds>
```