

Advanced Competitive Programming

國立成功大學ACM-ICPC程式競賽培訓隊
nckuacm@imslab.org

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Week 6

Sort Feast

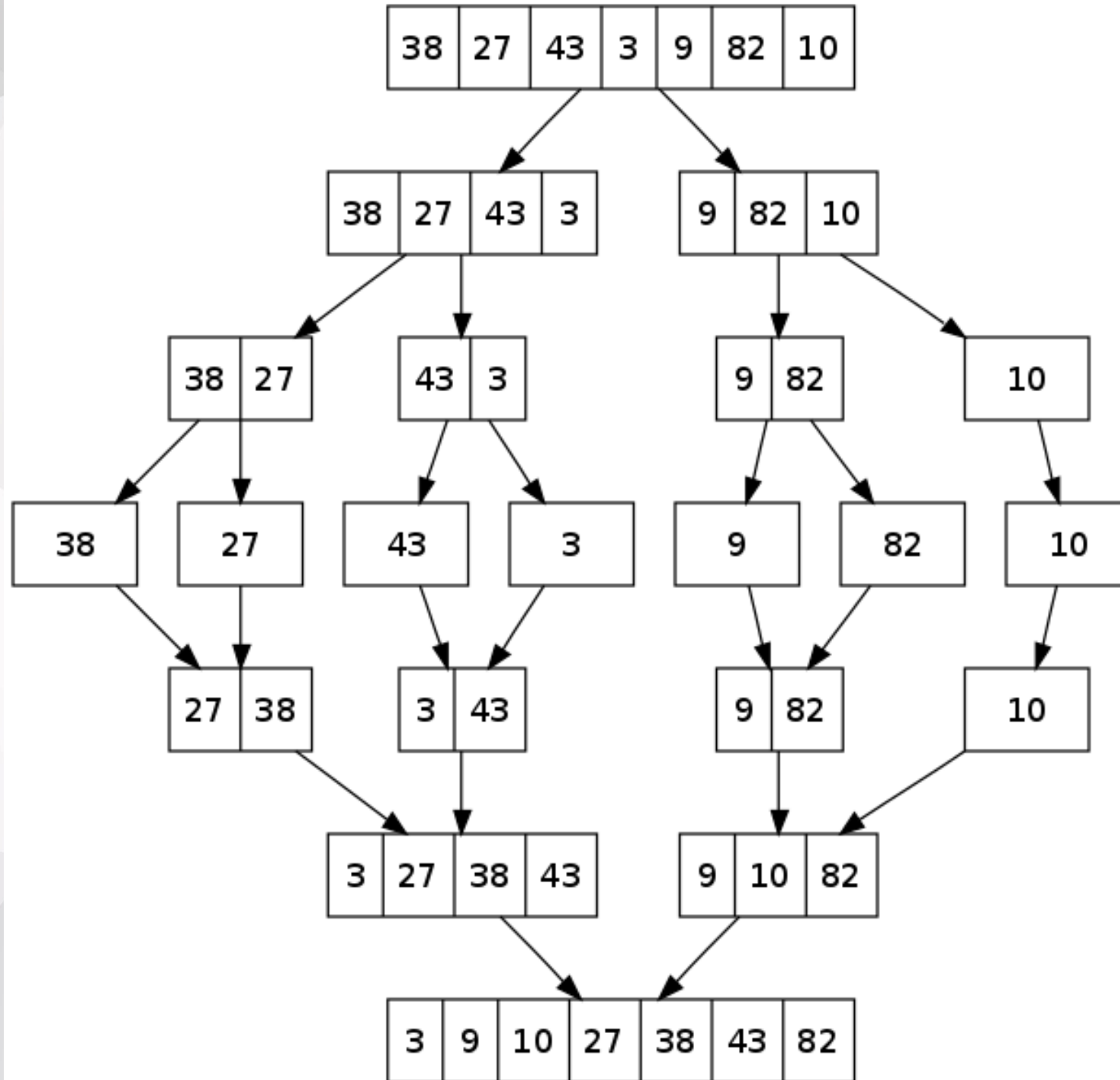
Outline

- Merge Sort
- Quick Sort
- Counting Sort

What about Bubble Sort?

Merge Sort

- 合併排序法
- 運用 Divide and Conquer
- $O(N \lg N)$



6 5 3 1 8 7 2 4

Merge Sort

```
void mergesort(int l, int r) { //[1, r)
    if (r-1 <= 1) return;
    int m = (l+r)/2;
    mergesort(l, m);
    mergesort(m, r);
    merge(l, r);
}
```


Quick Sort

- 快速排序法
- 運用 Divide and Conquer
- 大約 $N \lg N$ ，最差 N^2

Quick Sort

- 選定 pivot (作為一個比較的標準)
- 目標將數列中小於 pivot 的放到 pivot 的左邊，其餘在右邊
- 稱為 Partition

- 然後往 pivot 的左右遞迴排序

Quick Sort

```
int a[maxn];
int partition(int l, int r) {
    int p=a[r], ls=l; //p:pivot, ls:less equal
    for (int i = l; i < r; i++)
        if (a[i] <= p) swap(a[i], a[ls++]);
    swap(a[r], a[ls]);
    return ls;
}
```

Quick Sort

```
void quicksort(int l, int r) { //[l, r]
    if (l >= r) return;
    int s = partition(l, r);
    quicksort(l, s-1);
    quicksort(s+1, r);
}
```

Counting Sort

- 神奇的 $O(n)$
- 不用比較大小
- 不用儲存原數列
- 只適用於數字範圍不大的整數排序

- 計算出現過的數字的數量

Counting Sort

- 自己想怎麼寫

STL Sort is great.

Questions?
