Sorting Algorithms

- Number of Comparisons
- Number of Data Movements
- Additional Space Requirements

Sorting Algorithms

- Selection Sort
- Insertion Sort
- Bubble Sort
- Shaker Sort
- Merge Sort
- Heap Sort
- Quick Sort
- Bucket & Radix Sort
- Counting Sort

Animation Demos

http://www.cse.uta.edu/~holder/courses/cse3320/lectures/applets/sort1/heapsort.html

http://cg.sce.carleton.ca/~morin/misc/sortalg/
Stable Sort

- A sort is stable if equal elements appear in the same order in both the input and the output.
- Which sorts are stable? Homework!

Radix Sort

Algorithm
for i = 1 to d do
    sort array A on digit i using any sorting algorithm

Time Complexity: $O((N+m) + (N+m^2) + \ldots + (N+m^d))$

Space Complexity: $O(md)$
### Counting Sort

**Initial Array:**

```
1 2 3 4 5 6 7 8
2 5 3 0 2 3 0 3
```

**Counts:**

```
0 1 2 3 4 5
2 0 2 3 0 1
```

**Cumulative Counts:**

```
0 1 2 3 4 5
2 2 4 7 7 8
```

---

### External Sorting Methods

- **Assumptions:**
  - Data is too large to be held in main memory.
  - Data is read or written in blocks.
  - 1 or more external devices available for sorting.
- Sorting in main memory is cheap or free.
- Read/write costs are the dominant cost.
- Wide variety of storage types and costs.
- No single strategy works for all cases.

---

### External Merge Sort

- **Initial distribution pass**
- **Several multi-way merging passes**
- With 2P external devices:
  - Space for M records in main memory.
  - Sorting N records needs $1 + \log_{2P}(N/M)$ passes.
- With 2P >= external devices:
  - Sorts N records in $1 + \log_{2P}(N/M)$ passes.
### Order Statistics

- **Maximum, Minimum**  
  \( n-1 \) comparisons
  
  \[
  \begin{array}{c}
  7 \ 3 \ 1 \ 9 \ 4 \ 8 \ 2 \ 5 \ 0 \ 6
  \end{array}
  \]

- **MinMax**  
  - \( 2(n-1) \) comparisons  
  - \( 3n/2 \) comparisons

- **Max and 2ndMax**  
  - \( (n-1) \times (n-2) \) comparisons  
  - ???