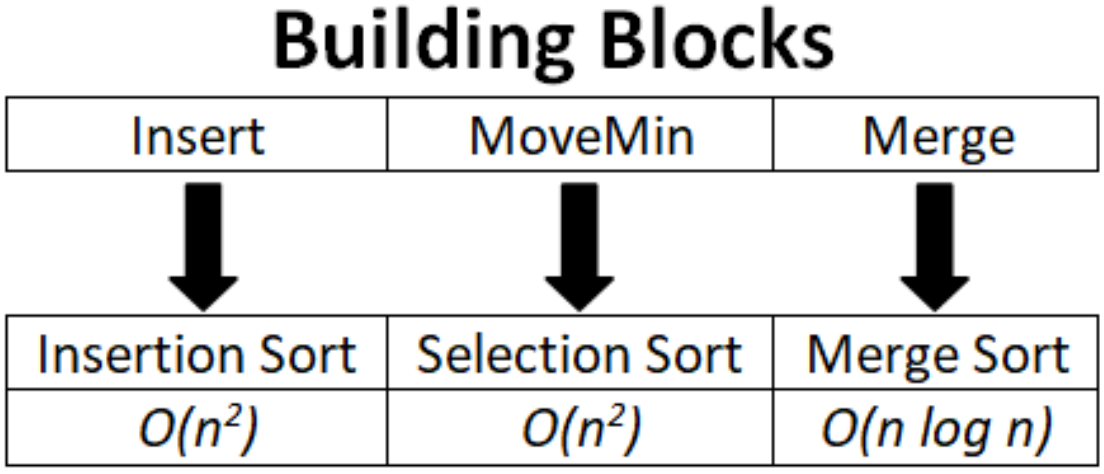
So far...


Building Blocks
Insert
MoveMin
Merge
Insertion Sort
Selection Sort
Merge Sort
O(n²)
O(n²)
O(n log n)

Why do we need sorting?

- Searching in sorted array vs unsorted array
- Finding a closest pair in n numbers
- Finding duplicates in the data
- Determining frequency distribution
- What is the $k-$th largest element in the data?

Characteristics of a Nice Sorting Algorithm

A decent sorting algorithms should be:

- Accurate
- Efficient
- Stable (take care of the satellite data)
- In-place

Permutations

- How many possible permutations are there for a sequence of n numbers?
- In sorting, we are interested in a specific permutation such that:

  $a_1 < a_2 < a_3 < ... < a_n$
- Crude way to search for the desired permutation out of all possible permutations will take $- - --$ time?

Divide / Partition

Given an input array $A[1..16]$ as given below:
503,87,512,61,908,170,897,275,653,426,154,509,612,677,765,703

The output should look like:
87,61,170,275,426,154,503,512,908,897,653,509,612,677,765,703
87,61,170,275,426,154,503,512,908,897,653,509,612,677,765,703

- Pivot / key
- Please write an efficient building block to do partitioning
- How can we use this building block to do sorting?
- Complexity?

Quick Sort

- Worst case complexity of quick sort?
- Best case complexity of quick sort?
- Please figure out exact scenarios for both of the above cases
- Stable?
- In-place?

Problems

Confusion

Does the complexity of a sorting algorithm change if we shift from ascending sort to descending sort?

Adding to Confusion

Given an unsorted array A of n numbers, what would be cost of answering following question:

1. Find the pair $x, y \in A$ (arbitrary) that maximizes $|x - y|$.
2. Find the pair $x, y \in A$ (arbitrary) that minimizes $|x - y|$.
3. Find the pair $x, y \in A$ (sorted) that maximizes $|x - y|$.
4. Find the pair $x, y \in A$ (sorted) that minimizes $|x - y|$.

More Problems

1. What is **average (mean)** of n numbers? How to compute it? Complexity?
2. What is **mode** of n numbers? How to compute it? Complexity?
3. What is **median** of n numbers? How to compute it? Complexity?