

Data Structures and Algorithms

Agenda

- Recursive Binary Search
- Sorting Algorithms
 - Selection sort
 - Bubble sort
 - Insertion sort

Q&A: Recursion vs Backtracking

- Recursion: Calling function itself.
- Recursion applications:
 - Divide and conquer: e.g. Binary search, Quick sort, Merge sort, ...
 - Simple: e.g. Factorial, Power, ...
 - Backtracking: Use recursion to check all possible options. e.g. 8-queen problem, ...

Sorting Algorithms

- Sorting: Arranging elements in ascending or descending order.
- Algorithms
 - Selection sort
 - Bubble sort
 - Insertion sort
 - Quick sort
 - Merge sort
 - Heap sort

Selection sort

Bubble sort

Insertion sort

Sorting

Stable sort vs Unstable sort

- Array: [{A, 65}, {B, 90}, {C, 55}, {D, 85}, {E, 55}, {F, 65}]
- Stable sort:
 - Equal elements maintains their relative order as in original array -- Guaranteed.
 - [{C, 55}, {E, 55}, {A, 65}, {F, 65}, {D, 85}, {B, 90}]
 - e.g. Bubble, Insertion, ...
- UnStable sort:
 - Equal elements may not maintain their relative order as in original array.

- [{C, 55}, {E, 55}, {F, 65}, {A, 65}, {D, 85}, {B, 90}]
- e.g. Selection.

In-place sort vs Out-place sort

- In-place sort
 - No additional space requires for holding array element.
 - Aux Space complexity is $O(1)$
 - e.g. Selection, Bubble, Insertion, ...
- Out-place sort
 - Additional space requires for holding sorted array element.
 - Aux Space complexity is $O(n)$ -- without stack space.
 - e.g. Merge.