tags: - cocubes - technical - algorithms

Design and Analysis of Algorithms

Big-O vocabulary

- O: upper growth bound, commonly used for worst-case complexity.
- Omega: lower growth bound.
- Theta: tight bound.
- Ignore constant factors and lower-order terms for large input size.

Growth order:

$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$.

Search

Algorithm	Requirement	Time
Linear search	none	O(n)
Binary search	sorted, indexable sequence	O(log n)

Binary search repeatedly removes half the search range. It is a divide-and-conquer algorithm.

Sorting table

Sort	Best	Average	Worst	Stable?	Extra memory
Bubble, optimised	O(n)	$O(n^2)$	$O(n^2)$	yes	O(1)
Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$	normally no	O(1)
Insertion	O(n)	$O(n^2)$	$O(n^2)$	yes	O(1)
Merge	O(n log n)	O(n log n)	O(n log n)	yes	O(n)
Quick	O(n log n)	O(n log n)	$O(n^2)$	normally no	O(log n) stack average
Heap	O(n log n)	O(n log n)	O(n log n)	no	O(1)

Fast selection rules:

- Nearly sorted small input: insertion sort.
- Linked list: merge sort, because it does not need random indexing.
- Guaranteed O(n log n): merge or heap sort.
- Typical fast in-memory array sort: quicksort, but worst case is $O(n^2)$.

Recursion

Every recursive algorithm needs:

1. base case;
2. progress toward the base case;
3. recursive call.

Example: factorial(n) = n × factorial(n−1), with factorial(0) = 1.

Watch for stack overflow if the depth is large.

Divide and conquer

Divide problem into smaller subproblems, solve recursively, combine.

Examples: binary search, merge sort, quicksort.

Greedy

Make the best local choice at each step. It works only when the problem has the required greedy-choice property.

Examples often tested: activity selection, Kruskal/Prim for MST, Dijkstra with nonnegative weights.

Dynamic programming

Store answers to overlapping subproblems.

- Memoization: top-down recursion plus cache.
- Tabulation: bottom-up table.

Use when subproblems overlap; do not call every recursion problem dynamic programming.

Graph algorithms

Algorithm	Main use	Key condition
BFS	shortest path in unweighted graph	queue
DFS	traversal/connectivity/cycle patterns	stack/recursion
Dijkstra	single-source shortest path	no negative edge weight
Bellman-Ford	allows negative edges	detects reachable negative cycle
Prim/Kruskal	minimum spanning tree	connected weighted undirected graph

Complexity tracing method

- Single loop to n: O(n).
- Nested independent loops to n: O(n^2).
- Loop doubling/halving i: O(log n).
- Loop inside n that halves: O(n log n).
- Consecutive blocks: add, then keep the largest term.

Example:

```
FOR i = 1 TO n
  j = 1
  WHILE j < n
    j = j * 2
```

Outer loop runs n times; inner loop runs log n times: O(n log n).

[[03_Technical/Data_Structures|Data structures]] · [[04_Pseudocode/Dry_Run_Method|Pseudocode dry-run]]