Advanced Algorithms and Data Structures Cheat Sheet

Contents

Algorithms and Data Structures Cheat Sheet

1 Graph Algorithms

Algorithm	**Problem Solved**	**Time Complexity**	**Notes**
DFS/BFS	Pathfinding, Connectivity	$O(V+E)$	DFS for cycles, BFS for shortest path in unweighted graphs.
Dijkstra	Single-source shortest path (SSSP)	$O((V+E)\log V)$	For graphs with non-negative weights.
Bellman-Ford	SSSP with negative weights	$O(VE)$	Detects negative weight cycles.
Floyd-Warshall	All-pairs shortest path	$O(V^3)$	Dynamic programming-based.
Kruskal/Prim	Minimum Spanning Tree (MST)	$O(E\log E)$	Use disjoint sets (Kruskal) or priority queues (Prim).
Edmonds-Karp	Max Flow	$O(VE^2)$	Uses BFS to find augmenting paths.

2 Dynamic Programming

Problem	**State Definition**	**Transition Formula**	**Complexity**
Knapsack	$dp[i][w]$: Max value with i items and weight w	$dp[i][w] = \max(dp[i-1][w], dp[i-1][w-wt[i]] + val[i])$	$O(nW)$
LCS	$dp[i][j]$: LCS of $A[0:i]$ and $B[0:j]$	$dp[i][j] = \max(dp[i-1][j], dp[i][j-1], dp[i-1][j-1]+1)$	$O(nm)$
Matrix Chain	$dp[i][j]$: Min cost to multiply matrices i to j	$dp[i][j] = \min_k(dp[i][k] + dp[k+1][j] + cost)$	$O(n^3)$

3 Data Structures

Data Structure	**Operations**	**Time Complexity**	**Notes**
Binary Search Tree	Insert, Search, Delete	$O(h)$	$h = \log n$ for balanced BSTs.
Segment Tree	Range queries, updates	$O(\log n)$	Supports lazy propagation.
Fenwick Tree (BIT)	Range sum, point update	$O(\log n)$	Easy implementation.
Hash Table	Insert, Search	$O(1)$	Average case.
Union-Find	Find, Union	$O(\alpha(n))$	Use path compression.

4 Randomized Algorithms

Algorithm	**Problem Solved**	**Time Complexity**	**Notes**
Randomized QuickSort	Sorting	$O(n\log n)$	$O(n^2)$ in worst case.
Monte Carlo Min-Cut	Minimum cut in graph	$O(n^2\log n)$	Success probability increases with retries.

5 Approximation Algorithms

Problem	Approximation Ratio	Algorithm	Notes
Vertex Cover	2	Greedy pick of edges	Select edges until all covered.
TSP (Metric)	1.5	MST-based	Approximation valid only for metric TSP.
Set Cover	$\ln n$	Greedy heuristic	n: number of sets.

6 Key Theorems

- **Max-Flow Min-Cut:** Max flow equals the capacity of the minimum cut.
- **Master Theorem:** Solves recurrences of the form $T(n) = aT(n/b) + O(n^d)$.
- **NP-Completeness:** If one NPC problem is in P, then $P = NP$.

7 Complexity Classes

Class	Definition	Example Problem
P	Solvable in polynomial time	Shortest Path
NP	Verifiable in polynomial time	SAT, HAM-CYCLE
NPC	Hardest problems in NP	3-CNF-SAT, TSP
$PSPACE$	Solvable using polynomial space	Quantified SAT

8 Formulas

- **Probability Rules:**
  - $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
  - $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$.
- **Big-O Growth Rates:**
  - $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$