Contents

Review Topics

Asymptotics

- Defining the asymptotic notations $\mathcal{O}, \Omega, \Theta$.
- Describing runtime/space complexity using asymptotic notations (best, expected, worst).

Recurrence

- Master's Theorem (3 cases).
- Recurrence Tree + Substitution Method.

Proof Techniques

- Combinatorial Argument.
- Induction (Weak/Strong).
- Contradiction.
- Bi-conditional (Equivalence) Proofs.

Heaps

- Definition of a heap.
- Properties of min/max heaps.
- Runtime of Build-Heap: $\mathcal{O}(n)$.

Binary Search Tree (BST)

Properties

- If y is in the left subtree of x, then key$[y] \leq$ key$[x]$.
- If y is in the right subtree of x, then key$[y] \geq$ key$[x]$.

Key Takeaways

- Worst-case height $h = \mathcal{O}(n)$; balanced BST: $h = \mathcal{O}(\log n)$.
- Min/Max/Search: $\mathcal{O}(h)$.
- Successor/Predecessor: $\mathcal{O}(h)$.

- Insert/Delete: $\mathcal{O}(h)$.
- Build-BST worst case: $\mathcal{O}(n^2)$.

Red Black Tree (RBT)

Properties

- Every node is red or black.
- Root is black.
- All leaves (NIL) are black.
- Red node → both children black.
- Every path to a leaf has the same black-height.

Key Takeaways

- Height $h = \mathcal{O}(\log n)$.
- Read-only ops same as BST.
- Rotation: $\mathcal{O}(1)$.

Sorting

Comparison-based

- Mergesort: $\mathcal{O}(n \log n)$.
- Heapsort: $\mathcal{O}(n \log n)$.
- Quicksort: $\mathcal{O}(n^2)$.
- Idea of Partition.
- Randomized Algorithms.
- Strong induction correctness proofs.

Non-comparison (finite key set)

- Counting sort: $\mathcal{O}(n + k)$.
- Radix sort: $\mathcal{O}(d(n + k))$.

Searching

- Searching in linear time (min, max, median, etc.).

Algorithm Design

- Start with a correct algorithm before optimizing.
  - Brute-force → sorting + binary search → data structures.
- Proof of correctness.
- Runtime + space complexity.
- A correct but suboptimal algorithm earns marks; incorrect but fast earns none.

Dynamic Programming

- Optimal substructure.
- Overlapping subproblems.
- Recurrence + Memoization.

Greedy

- Local optimum → global optimum (if correct).
- Reduction to smaller subproblem.
- First Greedy choice property.
- Optimal substructure.

Hashing

Hash Tables

- Universe U, key set K, table size m.
- Hash function $h : U \to \{0, \dots, m-1\}$.

Good Hashing Scheme

- Simple uniform hashing.
- Good collision resolution.

Collision Resolution

- Chaining.
- Linear probing (clustering issues).

Amortized Analysis

- Definition of amortized runtime.
- Aggregate analysis.
- Accounting method: credits + invariants.
- Relationship between charged price, credits, and actual cost.

Graphs

- Terminology: vertices, edges, paths, cycles, etc.
- Basic graph proofs.

Elementary Graph Algorithms

- BFS: $\mathcal{O}(V + E)$
- DFS: $\mathcal{O}(V + E)$
- Topological Sort: $\mathcal{O}(V + E)$

Minimum Spanning Trees

- Prim: $\mathcal{O}(E \log V)$.
- Key MST proof: adding an edge to MST creates cycle.

Single-Source Shortest Paths

- Relaxation: $\mathcal{O}(1)$.
- Bellman-Ford: $\mathcal{O}(VE)$ (detects negative cycles).
- Dijkstra: $\mathcal{O}((V + E) \log V)$ (no negative edges).
- Difference Constraints → BF solves feasibility.

Max Flow

- Flow and residual networks.
- Max-Flow–Min-Cut theorem.
- Ford-Fulkerson: $\mathcal{O}(E|f^*|)$.
- Edmonds–Karp: $\mathcal{O}(VE^2)$.
- Ford-Fulkerson + BFS.

Complexity Theory

- Definitions of P, NP, NPC.
- To prove $L \in$ NPC:
  1. Prove $L \in$ NP
     - Certificate (poly size)
     - Verification (poly)
  2. Pick known $L' \in$ NPC, prove $L' \leq_p L$
     - Polynomial-time reduction
     - Equivalence: $x \in L' \Leftrightarrow \alpha(x) \in L$

Exam Examples

Shortest Path (345 Final 2022)

m roads, n intersections. Roads are either country or main, directed, same length. Find shortest $s \to t$ path that uses only country roads first, then only main roads.

Amortized Analysis (345 Final 2019)

Binary min-heap with Insert/DeleteMin each $\mathcal{O}(\log n)$.
New op: Delete(k) deletes k smallest elements.
Use accounting to show: - Insert amortized $\mathcal{O}(\log n)$. - Delete(k) amortized $\mathcal{O}(k)$.

Max Flow Update (345 Final 2019)

Given max flow in G: - Increase capacity of one edge by 1 → update max flow in $\mathcal{O}(V + E)$. - Decrease capacity of one edge by 1 → update max flow in $\mathcal{O}(V + E)$.

Complexity Theory: 4-CNF-NAE-SAT (345 Final 2019)

Given 4-CNF formula Φ, NAE means each clause must have ≥1 true and ≥1 false literal.

Tasks: - Prove in NP
- Certificate
- Verification
- Reduction: 3-CNF-SAT $\leq_p$ 4-CNF-NAE-SAT
- Transformation
- Equivalence proof

Greedy: Minimizing Art Installations (345 Final 2018)

Given $b_1 \leq ... \leq b_n$, pick $A = \{a_1, ..., a_m\}$ so each b_i is within 250 of some a_j, minimizing $|A|$.

Tasks: - Devise greedy algorithm. - Prove greedy-choice property.

Greedy: Maximizing Course Satisfaction (358 Final 2019)

Given bloatness b_i and flavor f_i, maximize:

$$S(O) = \sum_{i=1}^{n} \left(f_{o_i} \left(1 - \sum_{j=1}^{i} b_{o_j} \right) \right)$$

Tasks: - Devise greedy algorithm. - Prove greedy-choice property.