# Algorithm Complexity & Operations Cheat Sheet

## 1. Union-Find (Disjoint Set) Variations (Week 7)

| Variant | Complexity (M ops) | Tree Height | Key Mechanism |
|---|---|---|---|
| Quick Find | O(MN) | Flat (<= 1) | Find is O(1), Union is O(N). |
| Quick Union | O(MN) | Unbounded (<= N) | Lazy union. Find can take O(N). |
| Weighted Quick Union | O(M log N) | O(log N) | Link smaller tree to larger tree. |
| Weighted + Path Comp. | O(M log* N) ~ O(M) | ~ O(1) | Flatten path during find. |

## 2. Sorting Algorithms (Weeks 1-3)

**Insertion Sort**
- Best Case (Sorted): O(n) time. n-1 comparisons, 0 swaps.
- Worst Case (Reverse): O(n^2) time. ~n^2/2 comparisons & swaps.
- Average Case: O(n^2) time. ~n^2/4 comparisons & swaps.

**Merge Sort**
- Complexity: O(n log n) all cases.
- Comparisons: <= n log n. Exact worst case: n log n - n + 1.
- Space: O(n) (Not in-place).

**Quick Sort**
- Best Case: O(n log n) (Pivot is median).
- Worst Case: O(n^2) (Pivot is min/max). Comparisons: n(n-1)/2.
- Average Case: O(n log n).

**Heap Sort**
- Heap Construction: O(n).
- Sorting Phase: n * O(log n).
- Total: O(n log n) all cases.

## 3. String Matching (Week 12)

| Algorithm | Best Case | Worst Case | Notes |
|---|---|---|---|
| Brute Force | O(n) | O(mn) | Comps in worst case: m(n-m+1). |
| Rabin-Karp | O(n+m) | Theta((n-m+1)m) | Worst case: Spurious hits (collisions). |
| Boyer-Moore | O(n/m) | O(mn) | Best: Skips m chars. Worst: Pattern matches suffix, fails front. |

## 4. Graph Algorithms (Weeks 4-6)

| Algorithm | Time Complexity | Key Detail |
|---|---|---|
| BFS / DFS | O(V + E) | Linear (Adj List). |
| Topological Sort | O(V + E) | Only for DAGs. Uses DFS or Kahn's. |
| Dijkstra's | O(E log V) | Greedy (Priority Queue). No negative weights. |
| Prim's (MST) | O(E log V) | Greedy. Grows tree from a node. |
| Kruskal's (MST) | O(E log E) | Greedy. Sorts edges. Uses Union-Find. |

## 5. Dynamic Programming & Greedy

# Algorithm Complexity & Operations Cheat Sheet

- **Coin Change (Greedy): O(n). Optimal only for canonical systems.**
- **Coin Change (DP): O(S * n). S=target, n=denominations.**
- **LCS (DP): O(nm). Space O(nm).**
- **0/1 Knapsack (DP): O(nW). Pseudo-polynomial.**
- **Matrix Chain Multiplication: Time O(n^3), Space O(n^2).**