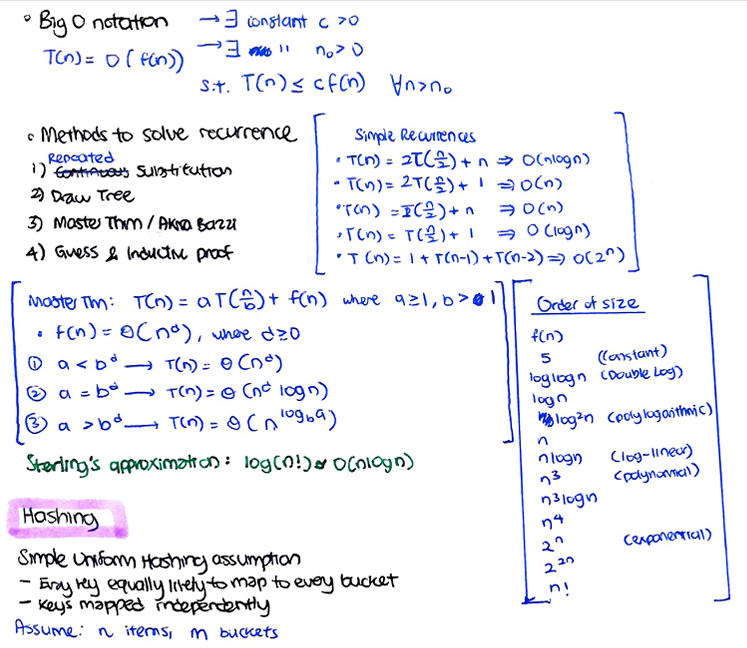
Sorting Algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Best Case | Avg Case | Worst Case | Extra Memory | Stable |
| Bubble Sort | O(n) | O(n^2) | O(n^2) | O(1) | Yes |
| Selection Sort | O(n^2) | O(n^2) | O(n^2) | O(1) | No |
| Insertion Sort | O(n) | O(n^2) | O(n^2) | O(1) | Yes |
| Merge Sort | O(nlogn) | O(nlogn) | O(nlogn) | O(n) | Yes |
| Quick Sort | O(nlogn) | O(nlogn) | O(nlogn | O(1) | No |



Dijkstra’s Running Time with different Implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PQ Implementation | Insert | deleteMin | decreaseKey | Total |
| Array | 1 | V | 1 |  |
| AVL Tree | log V | log V | log V | O(E log V) |
| d-way Heap |  |  |  |  |
| Fibonacci Heap | 1 | log V | 1 | O(E + V log V) |

Shortest Path’s Summary (SSSP)

|  |  |  |  |
| --- | --- | --- | --- |
| GRAPH | ALGO | TIME | DETAILS |
| No negative weight cycles | Bellman-Ford | O(VE) | Early termination if sequence of |E| relax op no effect - can detect negative weight cycle |V| loop relax still has effect |
| No negative edges (cos proof by contradiction depends on it) | Dijkstra  //like BFS/DFS but w priorityQ | O(E log V) | Uses AVL Tree & Hashtable (get nodes) Running Time: O((V+E)log V) = O(E Log V)  - insert(key, priority): O(logn)  - deleteMin(): O(logn)  - decreaseKey(key, priority): O(logn)  - contains(key): O(1) |
| No directed cycles (only acyclic graphs) | One pass Bellman Ford  (TopoSort + Relax)  //DFS Post-order to get Topological Ordering | O(V+E) = O(E) |  |
| Kahn’s Algorithm  //TopoSort as well | O(V+E) = O(E) |  |
| No cycles (undirected)  -> Undirected Trees (cos one node can only have one parent) | BFS/DFS + Relax  //cos only one way to path, will always be topo-order | O(V + E) = O(V) //no cycles E < V |  |
| Unweighted graph | BFS | O(V+E) |  |
|  |  |  |  |

