|  |  |  |
| --- | --- | --- |
| **Recursion**  Laws of Recursion:  1. A Recursive algorithm must call itself recursively  2. A Recursive algorithm must have a base case  3. A Recursive algorithm must change its state to move forward.  Types of Recursion  Tail – A recursive function in which the last statement is a recursive call  Head– A recursive function in which is executed before the statements  Tree – A recursive function that has more than 1 recursive call, which results in the function branching into a tree  **Backtracking**  3 Keys to Backtracking:  **Choice**: What is the sub problem or recursive function  **Constraints:** The rules for the base case  **Goal**: Goal is always to reach base case  **Heap**  Max heap  Insertion rules:  1. Make sure heap is a complete binary tree  2. Try fill the left side of the tree  3. Parent value must be greater than child  **insertion** adjust tree bottom to top  **deletion** adjust tree top to bottom    Min heap | **Trees**  Binary Tree  Min Nodes= h+1  Max Nodes=2power(h-1)    AVL  balance factor= height of left sub tree - height of right sub tree  Balance Factor=HL - HR ={-1,0,1}  Balance Factor = | hl -hr| <= 1  Don't Count Nodes. Count levels !!  Balance Factor is counted on every node!!  Single rotations: LL,RR  Double rotations: RL,LR    Btree |  |

|  |  |
| --- | --- |
| **Graphs**  Undirected Graph  Directed Adjacency Matrix  Weighted Graph | BFS and DFS    Minimum Spanning Tree |

|  |  |
| --- | --- |
| **Time Complexity** | **Sorting**    Merge Sort      Quick Sort |