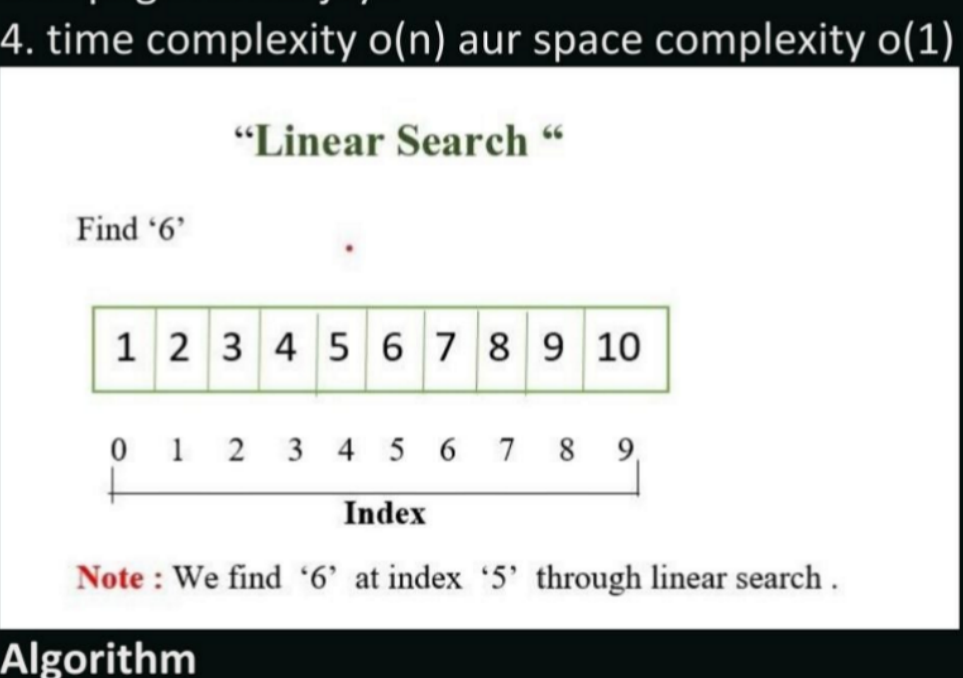
BCS 42 NOTES

Tuesday, February 6, 2024 8:06 PM

Linear search

1. basic aur simple search algorithm hai
2. array me ek ek element ko sequence me check karte hai aur jab target mil jata hai to use locate kar dete hai
3. ex. jese book me ek page ko dhundne k liye book k starting page se dhundte hai aur jab tak sahi page na mil jaye
4. time complexity o(n) aur space complexity o(1)



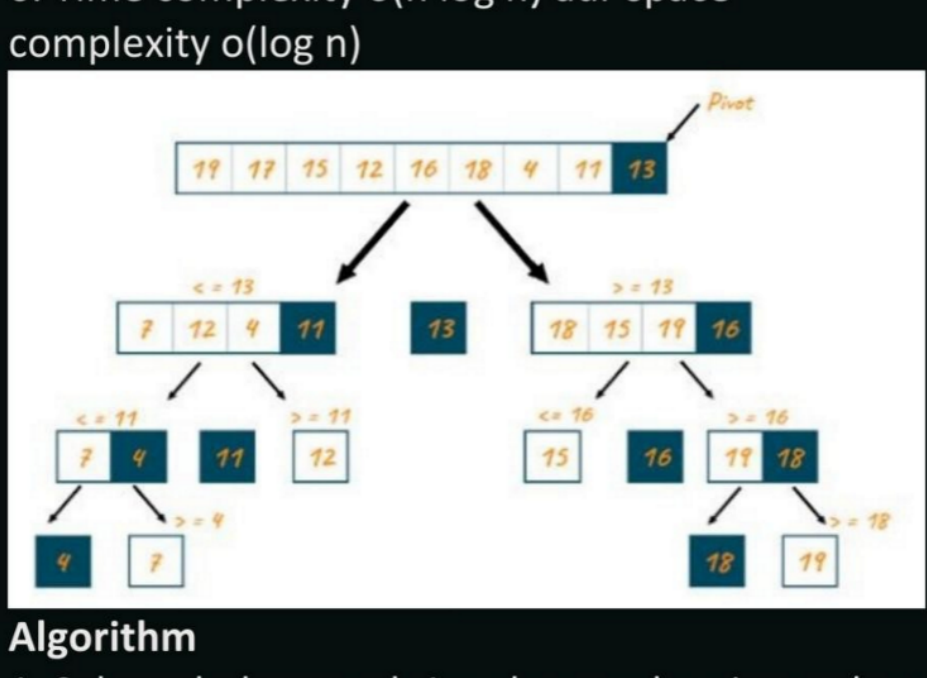
Note : We find '6' at index '5' through linear search .

Algorithm

1. sabse phele array k first element ko check kro
2. jo element ham dhund rahe kya wo equal hai to uska index return kardo
3. nhi hai to next element ko check karo
4. repeatedly poore array ko search karte raho
4. agar element nhi mila to element not found ko return kardo

Quick sort

1. ek sorting algorithm hai
2. divide and conquar approach par based hai
3. ek array ko chote parts me divide karte hai phir har part ko sort karte hai phir ihne merge karke sorted array banalete hai
4. programming language good for recursion
5. isme stability nhi hoti
6. Time complexity o(n log n) aur space complexity o(log n)



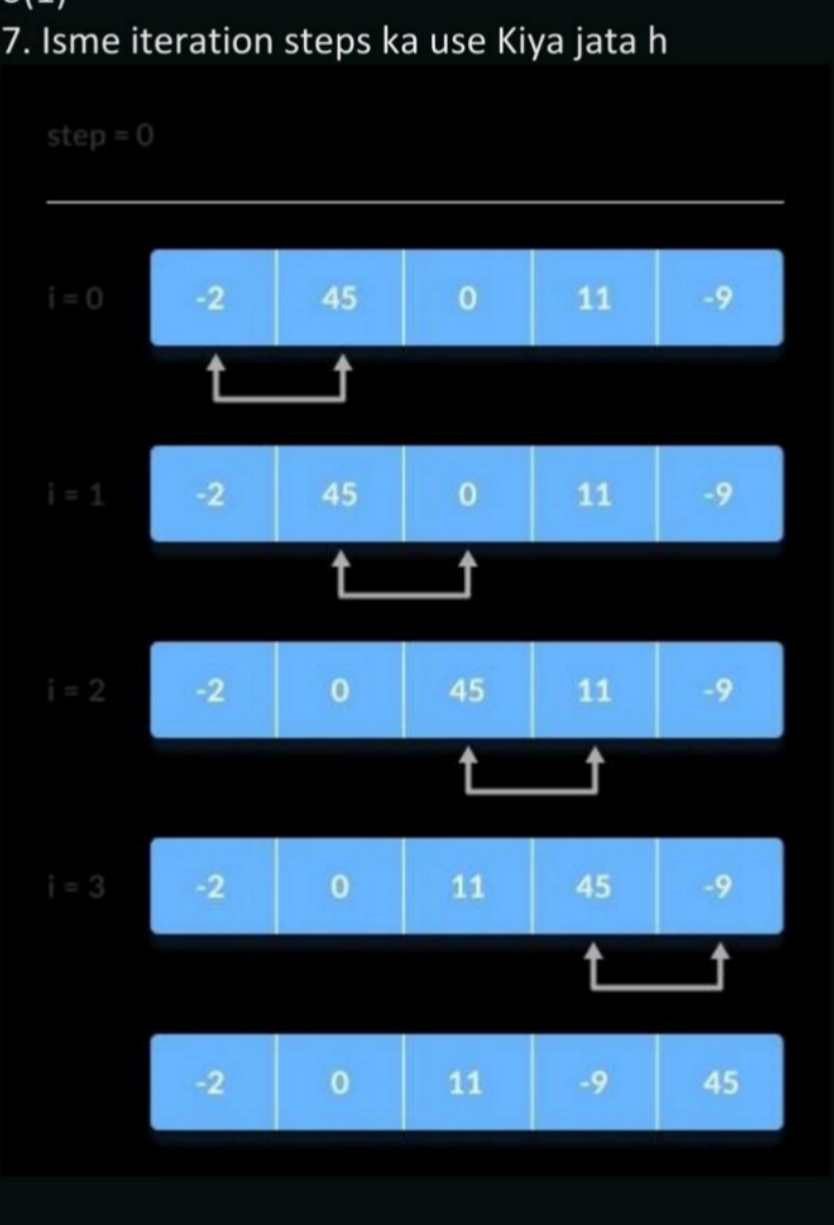
Algorithm

1. Sabse phele array k 1st element ko pivot select Karo
2. Left side me pivot se bada element search Karo aur usse p mark kardo aur right side me bhi pivot se chota element search Karo or use q as mark kardo
3. Mark kiye gaye p aur q ko ek dusre se swap kardo agar p aur q dono me cross ho jata h to q ko pivot se swap kardo jisse pivot array k andar aa jayega
4. Phir pivot element k left side aur right side me pivot ko chodkar step 1, 2, 3 baar baar repeat karte raho jab tak sare elements sort na ho jaye excluding jo phele se sorted h

Bubble sort best case is when array is already sorted
Quick sort worse is when array is in reverse order
Best case of bubble sort is same as the worse case of quick sort

Bubble sort

1. Bubble sort ek sorting algorithm hai
2. jisme adjacent elements ko compare kiya jata hai
3. aur agar wo galat order mein hain toh unhe swap kiya jata hai
4. Ye process chalte jata hai jab tak saare elements sahi order mein nahi aa jate.
5. Isme stability hoti h
6. Time complexity o(n) aur space complexity o(1)
7. Isme iteration steps ka use Kiya jata h



Algorithm

1. Sabse phele 1st iteration me 1st element ko 2nd element se compare Karo
2. Agar 1st element 2nd element se chota hai to kuch mat karo
3. Agar 1st element 2nd element se bada hai to swap kardo
4. Recursivly Next iteration me continue repeat karte raho jab tak saare elements sort na ho jaye

merge sort

1. merge ek popular sorting algorithm hai
2. iska principal divid and conquar algorithm hai
3. isme data ko do parts me divide karte hai phir each part ko recursively divide karo phir har individual part ko sort karo aur last me sorted parts ko merge karkar sorted single list banti hai
4. Isme stability hoti hai
5. time complexity o(n * log n) aur Space complexity o(n)

Advantages

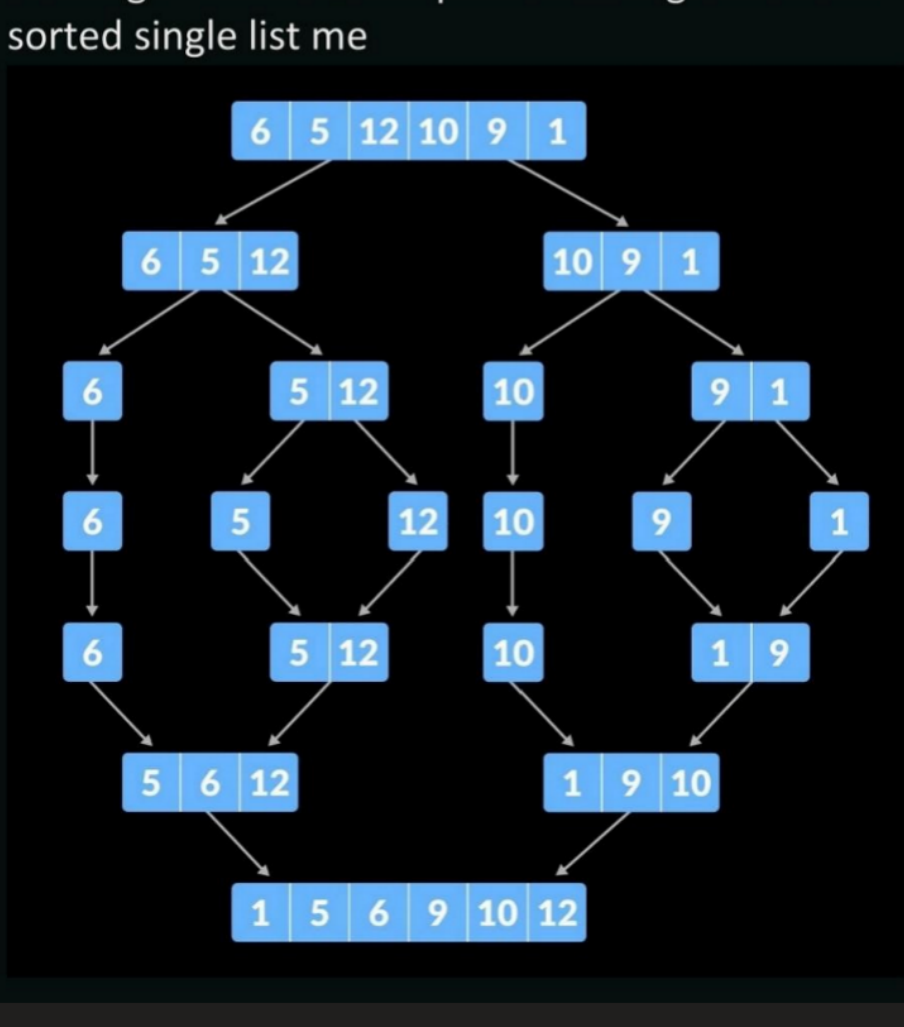
1. Sorted output : Merge sort hamesha sorted output deta hai
2. effiecient for large datasets
3. linked list k liye efficient

Disadvantage

complex for implementation
Small datasets k liye slow
worst case time complexity

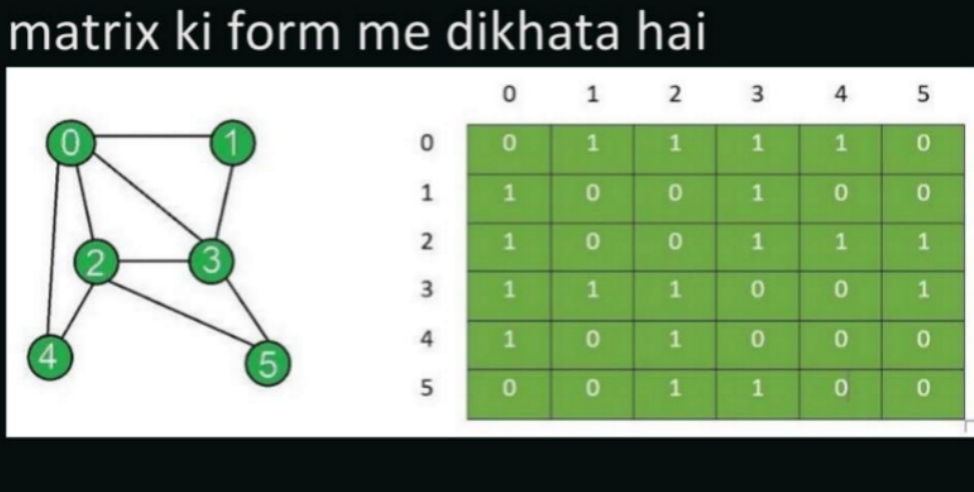
Algorithm

1. Divide - sabse phele unsorted list ko 2 parts me divide kardo
2. Conquer - each part ko recursively divide karo
3. phir individual part ko sort kardo
4. Merge - sabhi sorted parts ko merge kardo ek sorted single list me



Adjacency matrix

1. ek graph ko represent karne ka tareeka hai
2. jisme har node ko ek row aur column se represent kiya jata hai
3. Agar koi nodes ek dusre se connected hain, toh matrix mein uss spot par 1 hota hai, nahi toh 0.
4. Yeh hume graph ke connections ko ek matrix ki form me dikhata hai



	0	1	2	3	4	5
0	0	1	1	1	1	0
1	1	0	0	1	0	0
2	1	0	0	1	1	1
3	1	1	1	0	0	1
4	1	0	1	0	0	0
5	0	0	1	1	0	0

DFS depth first seach or depth first traversal

1. is recursive algorithm for searching all the vertics of the graph or tree data structure
2. traverse means visting all the nodes in the graph
3. ye stack data structure par perform hota hai
4. top

Algo.

1. sabse phele graph k kisi bhi ek vertex se stack k top par rakhkar start karo
2. stack ka top item lo or use visited list me add kardo
3. vertex k adjacent nodes ki list banao aur ek baar add karo jo visited list me nahi hai
3. step 2 aur 3 ko repeat karte raho jab tak stack empty hai

Dijkstra s Algorithm

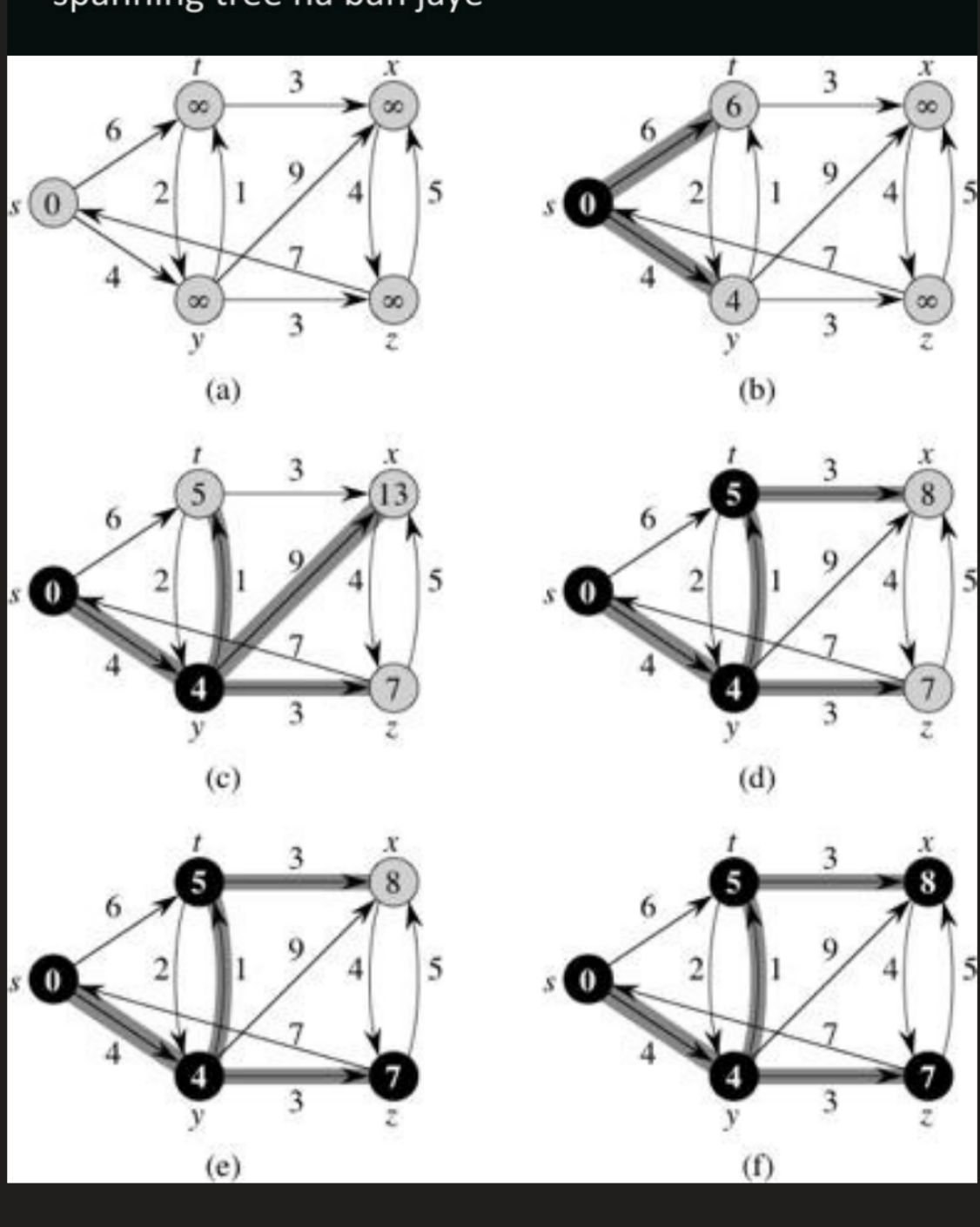
1. It differ from the minimum spanning tree because the shortest distance between two vertics might not include all the vertics of the graph
2. Time complexity o(e log v) aur space complexity o(v)

Applications

Game development, Circuit design, Robotics, Transportation, Find location in map

Algo

1. sabse phele agar loop or parellel path hai to remove kardo
2. ek starting vertex choose Karo or usme 0 assign Karo or uske near by me point karne wale baaki vertex me ∞ assign kardo
3. starting ka vertex ki value aur usse connected edge ka sum kardo point kiye vertex me
4. next vertex select Karo or uske point kiye gaye vertex me bhi vertex ki value or edge ka sum kardo agar ek vertex ko multiple vertex point karte hai to minimum edge vale k saath sum karo
5. ye procees repeat karte raho jab tak minimum spanning tree na ban jaye



Prims algorithm

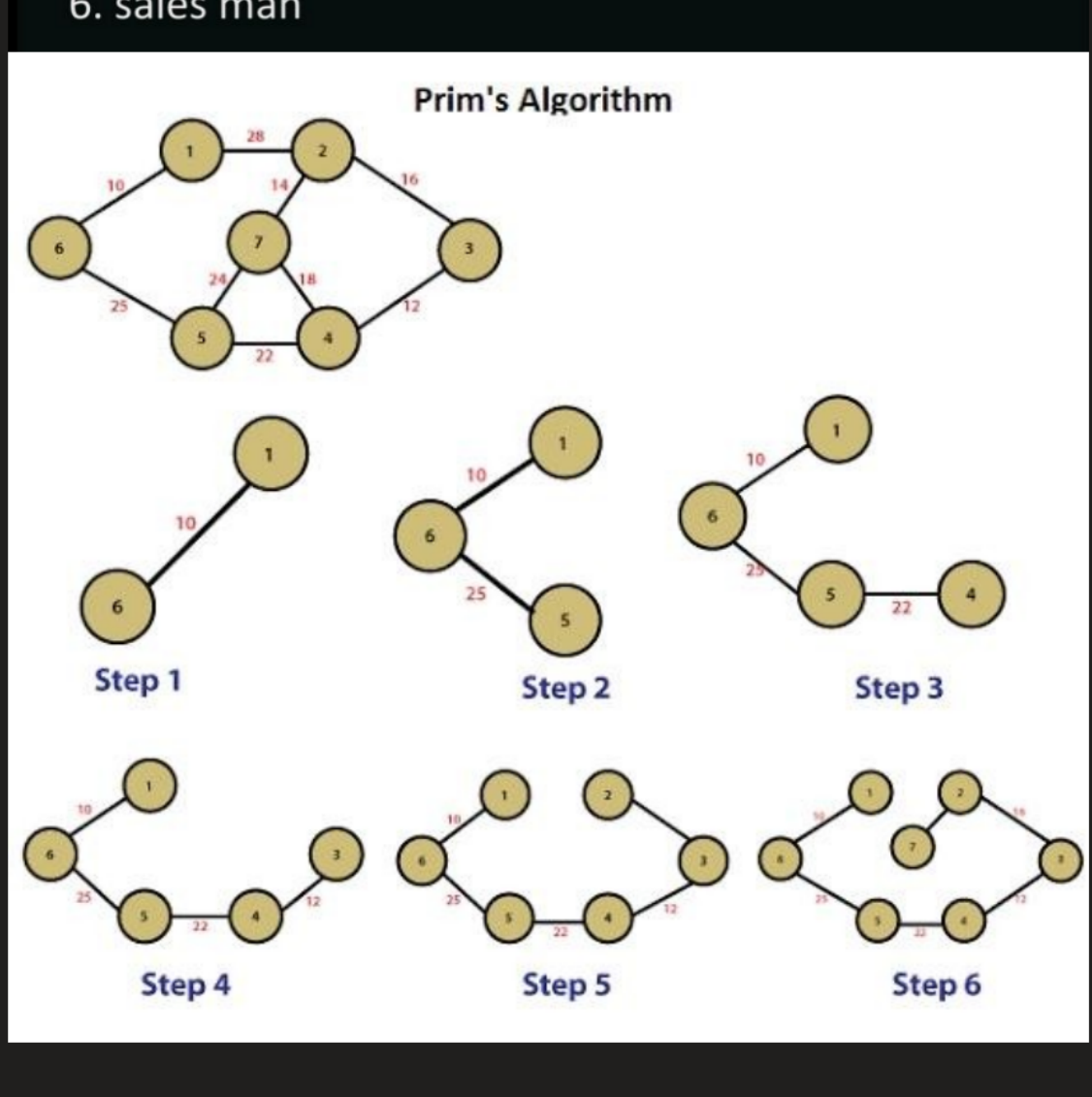
1. prism algo. Is a minimum spanning tree algo.
2. that takes a graph as an input and finds the subset of the edges of the graph
3. which form a tree that include every vertex
4. which has the minimum sum of weights amoung all the trees that can be formed from the graph
5. time compelexity o(e log v)

Algo

1. sabse phele ek random vertex choose karo or use tree ka root bana do
2. tree ko naye vertex se jodne wale minimum edge dhundo or use tree me shamil karlo yaad rahe repeated same edge bhi chalega or loop nhi banna chaiye
3. is process ko baar baar repeat karte raho jab tak minimum spanning tree na ban jaye

Applications

1. electrical wiring
2. traffic management
3. fraud detection
4. social media analysis
5. e commerse recommendation
6. sales man



Kruskal's Algorithm

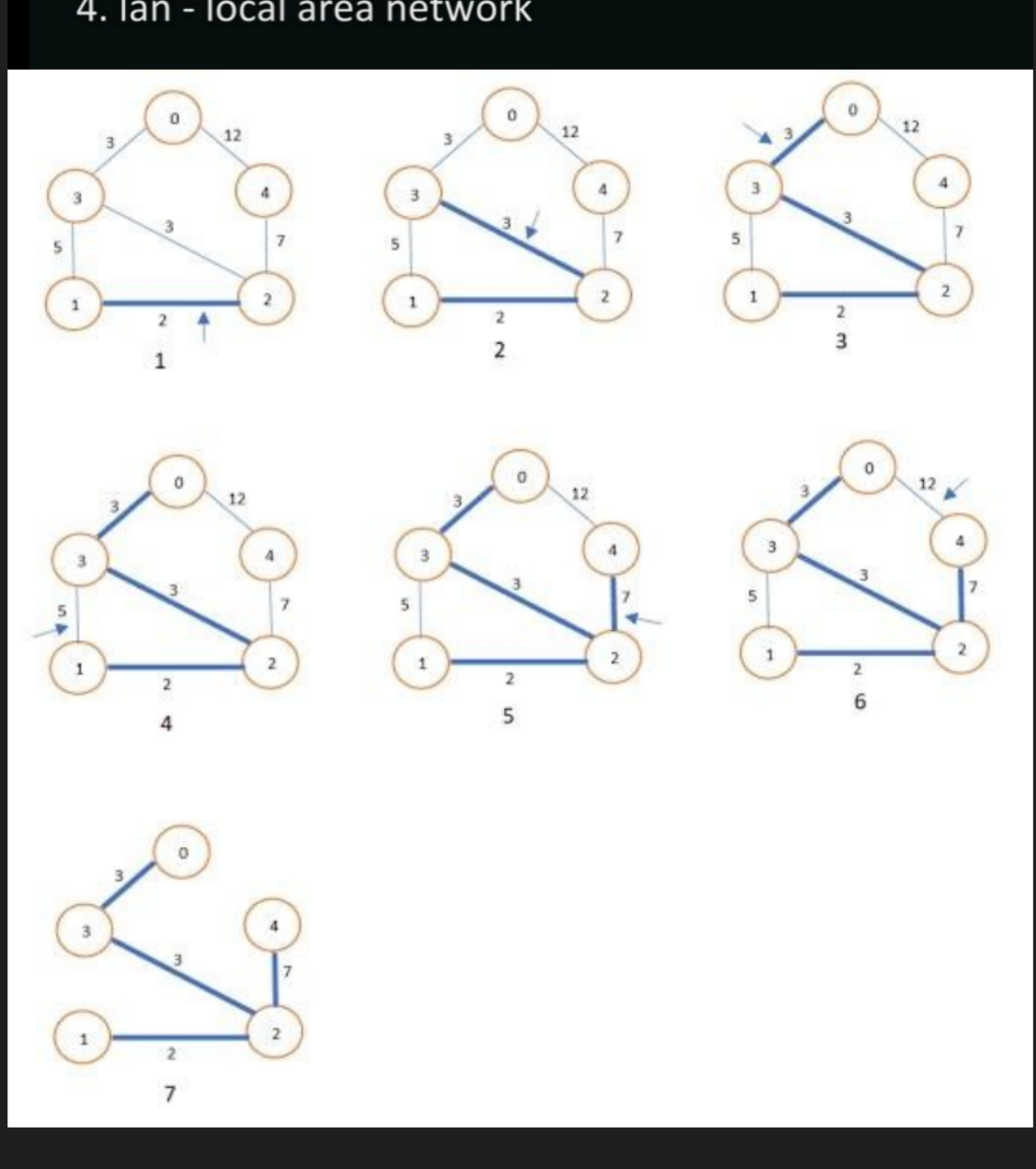
1. Kruksal algo. Is a minimum spanning tree algo.
2. that takes a graph as an input and finds the subset of the edges of that graph
3. which form a tree that include every vertex
4. which has the minimum sum of weights amoung all the trees that can be formed from the graph
5. time complexity o(e log e)

Algo

1. sabse phele saare edges ko low weight se hight weight ascending order me sort kardo
2. lowest minimum edge lo or use spanning tree me add kardo
3. Add kiya hua edge cycle nhi banna chaiye
4. is process ko repeatedly karte raho jab tak saare vertix nhi phoch jaaye

Applicatons

1. network design
2. circuit design
3. robotics
4. lan - local area network

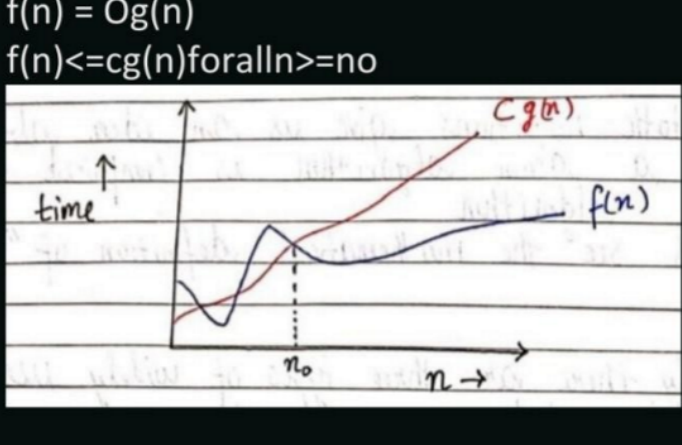


Asymptotic Notetimes

1. Asymptotic Notetimes give us an idea about how given algorithm is good compared to some other algorithm
2. Let us see the mathematical definition order of now, primary there are three types of widely used asymptotic :

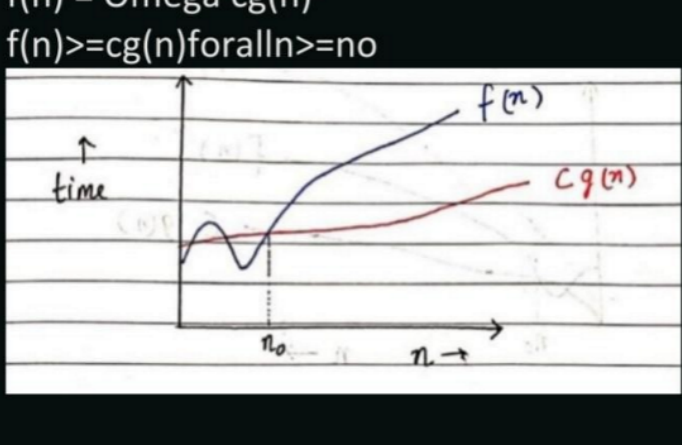
Big O

For Worst case
Big o notation is used to describe asymptotic upper bound
f(n) = Og(n)
f(n)<=cg(n)foralln>=no



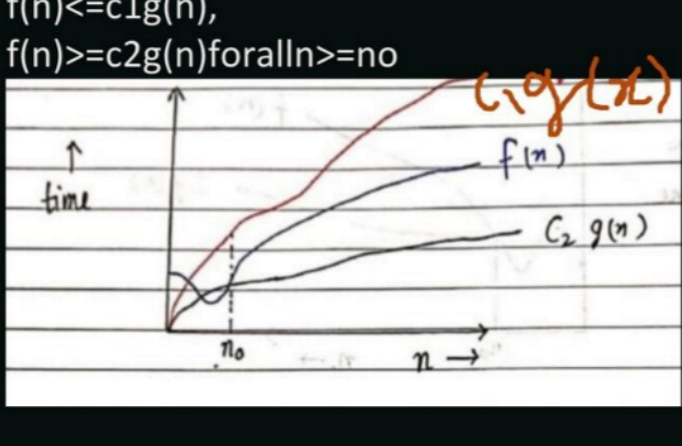
Big Omega

For Best case
Big Omega notation is used to describe asymptotic lower bound
f(n) = Omega cg(n)
f(n)>=cg(n)foralln>=no



Big Theta

For average case
Merge both equation and we get
f(n) = Theta cg(n)
f(n)<=c1g(n),
f(n)>=c2g(n)foralln>=no



Dynamic programming

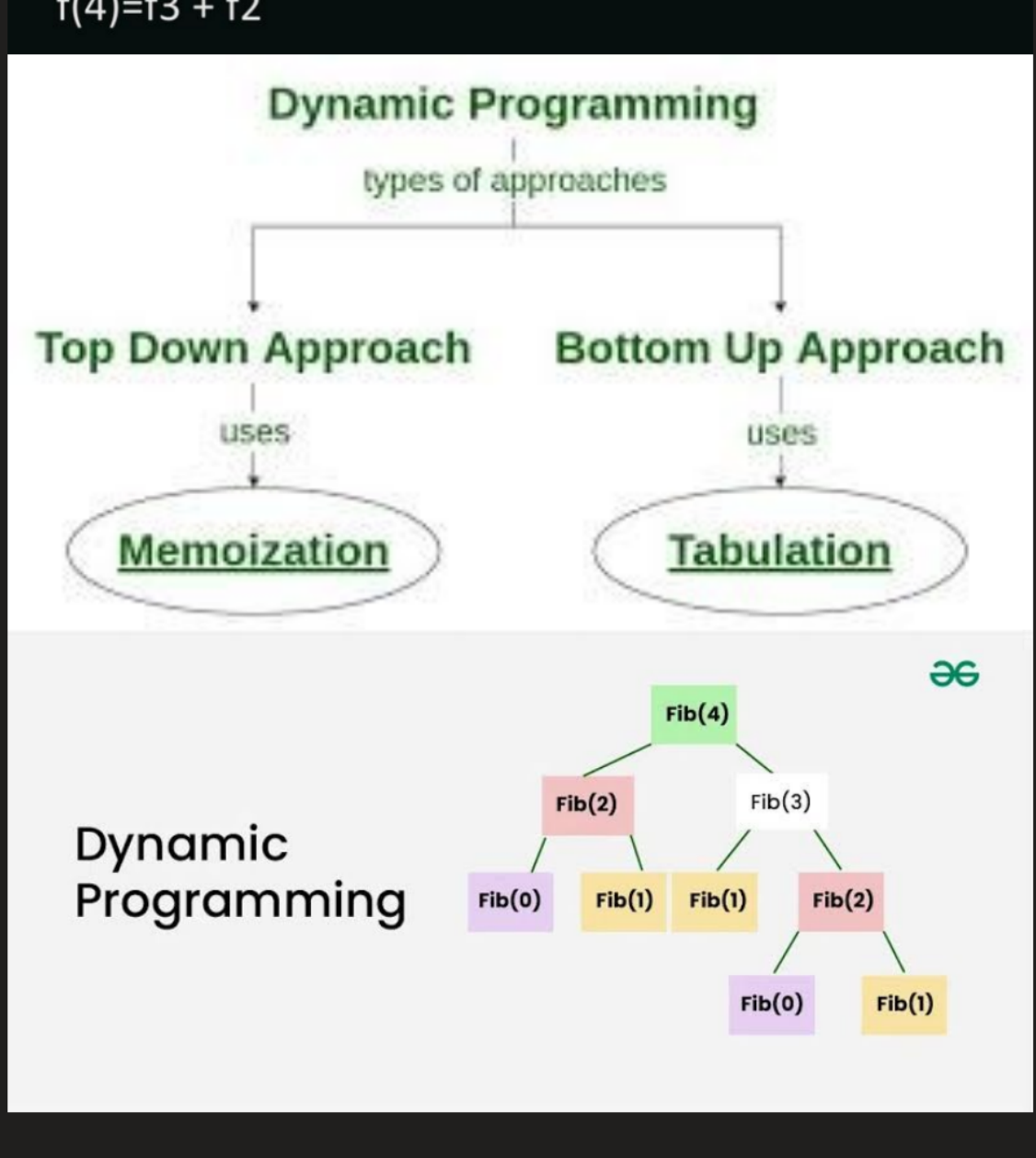
1. Ek problem solving approach h
2. Jisme big problems ko chote chote sub problems me divide karte h phir inke solutions ko store karke baad me reuse karte h
3. Is like a jigsaw puzzle by remember that how many piece you solved

Algorithm

1. First term is 0
2. Second term is 1
3. Third term is sum of step2 and step1
1+0=1
4. Fourth term is sum of step3 and step2
1+1=2
5. Fifth term is sum of step4 and step3
2+1=3

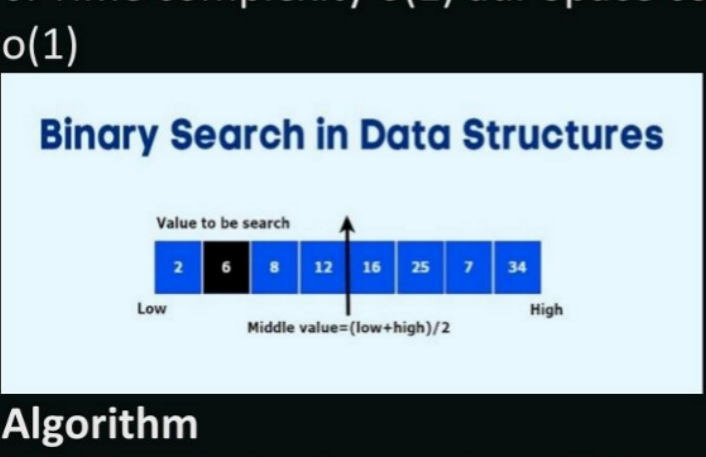
Peusudocode

```
f(0)=0
f(1)=1
f(2)=f1 + f0
f(3)=f2 + f1
f(4)=f3 + f2
```



Binary search

1. binary search divide or conquer technique hai
2. sorted list me middle me se element ko dhundna start karte hai agar element middle se chota hai to uski first half me dhundte hai
3. agar element middle se bada hai to uski second half me dhundte hai continuously ye steps repeat karte hai
4. ye fast or efficient hai
5. Ex. jese ek page ko dhundle k liye book ko middle me se kholte hai or agar wo page nhi milta to middle k left half or right half me search karte hai recursively
6. Time complexity o(1) aur Space complexity o(1)



Algorithm

1. Sabse phele element ko middle me se find karo
2. Agar element Target k equal h to index return kardo
3. agar element middle se chota hai to uski 1st half me dhundo
4. Agar element middle se bada hai to uski 2nd half me dhundo
5. Otherwise recursively search karte raho jab tak element nhi mil jata

BFS Breadth first search or breadth first traversal

1. is a recursive algorithm for searching all the vertics of the graph of the tree data structure
2. traversal means visiting all the nodes in the graph
3. the purpose of the algo. is to work each vertex as visited while avoiding cycles.
4. ye queue data structure par perform hota hai
5. back, front

algo.

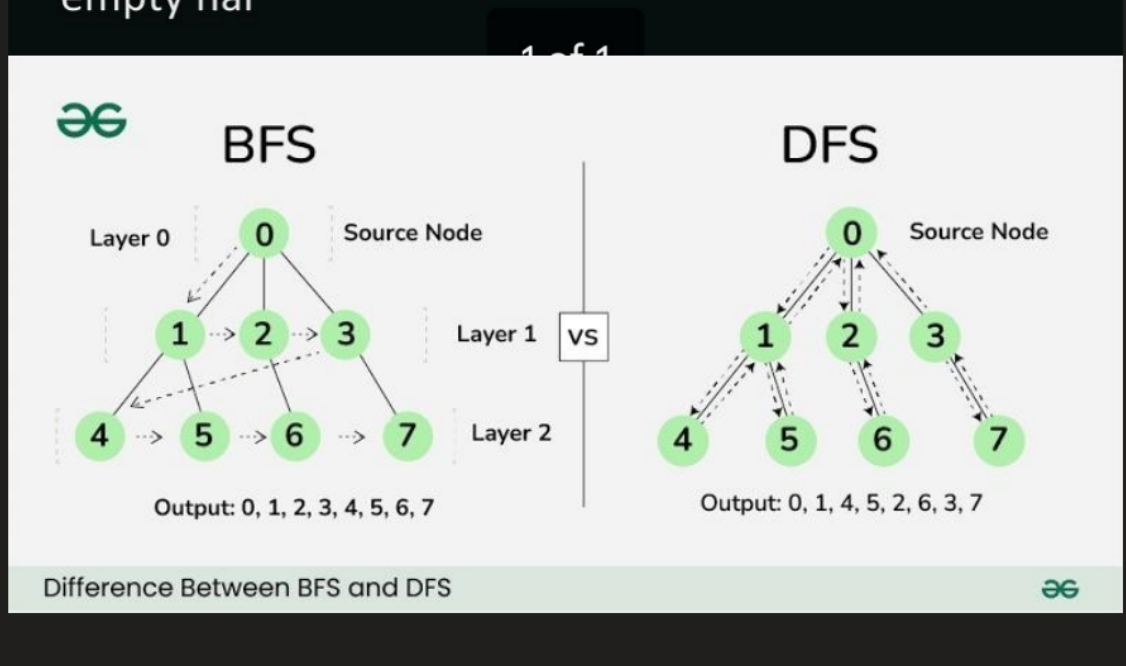
1. sabse phele graph k kisi bhi vertex se queue k piche se rakhkar start karo
2. queue ka front item ko aur use visited list me add kardo
3. vertex k adjacent nodes ki list banao aur ek baar add kardo jo visited list me nahi hai
4. step 2 and 3 ko repeat karte raho jab tak queue empty hai

DFS depth first seach or depth first traversal

1. is recursive algorithm for searching all the vertics of the graph or tree data structure
2. traverse means visting all the nodes in the graph
3. ye stack data structure par perform hota hai
4. top

Algo.

1. sabse phele graph k kisi bhi ek vertex se stack k top par rakhkar start karo
2. stack ka top item lo or use visited list me add kardo
3. vertex k adjacent nodes ki list banao aur ek baar add karo jo visited list me nahi hai
3. step 2 aur 3 ko repeat karte raho jab tak stack empty hai


BFS
Layer 0
Source Node
Layer 1
Layer 2
VS
DFS
Source Node
Output: 0, 1, 2, 3, 4, 5, 6, 7
Output: 0, 1, 4, 5, 2, 6, 3, 7
Difference Between BFS and DFS

By Ankit Kumar
Update 20 apr 24