Course: Advanced Algorithm and Problem Solving

WEEK 3 - Greedy Algorithms

Lemlem Kassa (Ph.D.)

Addis Ababa Science and Technology University, Ethiopia

April, 2025

Week 3: Greedy Algorithms

Content

1. Introduction to Greedy algorithm.
    - Characteristics of a Greedy Algorithm
    - How to use Greedy Algorithms
    - Advantages and Drawback of Greedy Approach
2. Type of Greedy Algorithms
    - Kruskal's Algorithm
    - Prim's Algorithm
    - Huffman Coding

Lecture Learning Outcome

- By the end of this lesson students will Understand:-
  - Greedy algorithm.
  - Application of Greedy Method
  - Characteristics of a Greedy Algorithm
  - How to use Greedy Algorithms
  - Advantages and drawback of Greedy Approach
  - Types of greedy algorithms -: Kruskal's, Prim's, and Huffman Coding.

1. Introduction to Greedy algorithm

What is Greedy Method?

- Greedy method is the most straight forward method and popular for obtaining the optimized solutions.
- **Optimization Problem**: the problem of finding the best solution (optimal solution) from all the feasible solutions (practicable of possible solutions).
- In an optimization problem we are given a set of constraints and an optimization functions.

- **Feasible solutions** : - Solutions that satisfy the constraints.
- **Optimal solution** :- the optimization function has the best possible value

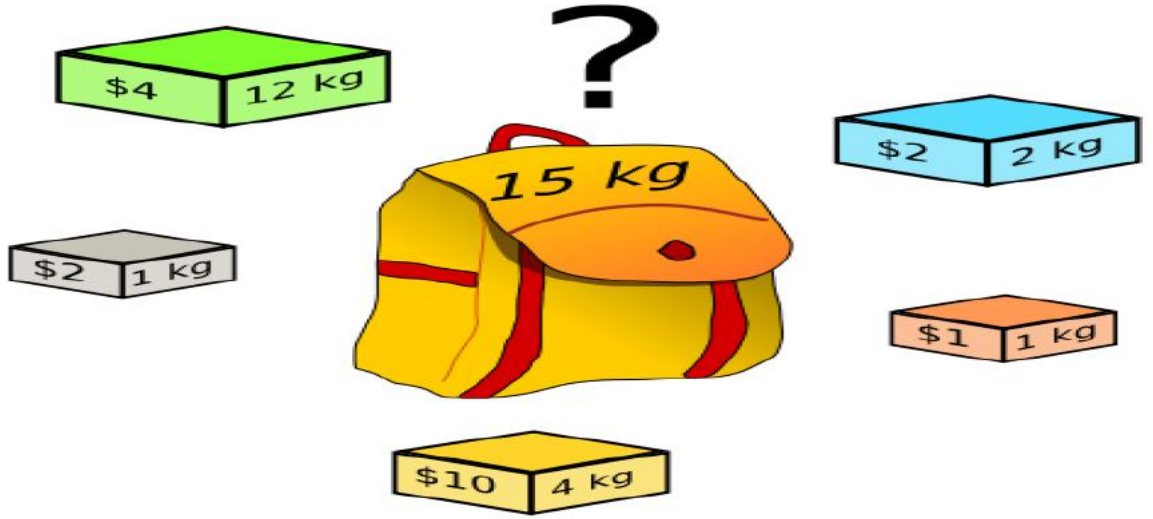
[1]. S.K. Sathua , M.R. Kabat, R. Mohanty. Design And Analysis Of Algorithms, Page 7. https://vssut.ac.in/lecture_notes/lecture1428551222.pdf

1. Introduction to Greedy algorithm …(cont'd)

Application of Greedy Method:-

Knapsack problem

- Given a set of items, each with a mass and a value, determine the number of each item to include in a collection.
- The total weight is less than or equal to a given limit and the total value is as large as possible



[1]. S.K. Sathua , M.R. Kabat, R. Mohanty. Design And Analysis Of Algorithms., Page 7. https://vssut.ac.in/lecture_notes/lecture1428551222.pdf

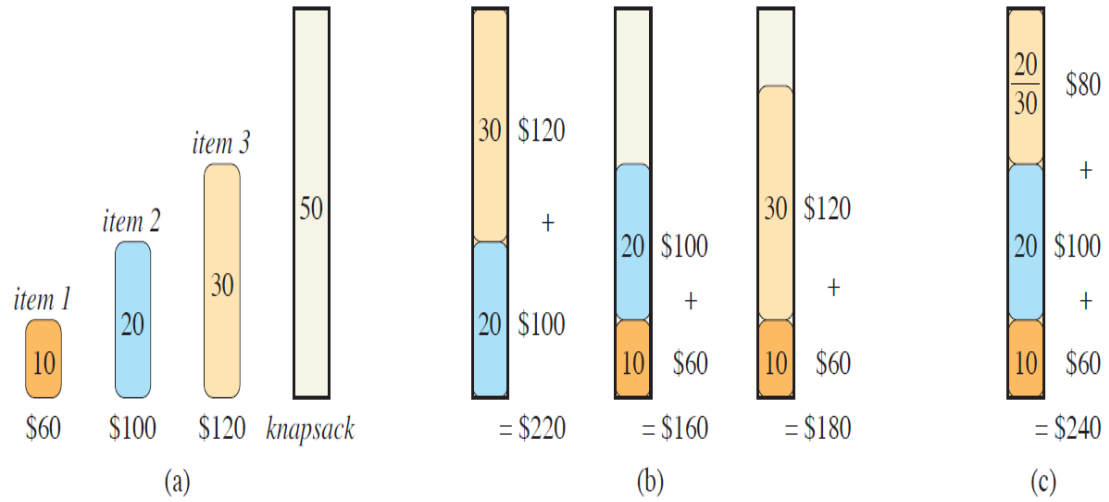
[2]. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. Page-430

1. Introduction to Greedy algorithm …(cont'd)

Two versions of the knapsack problem

1. 0/1 knapsack problem

- n items.
- Item i is worth $ $_i$, weighs w_i pounds.
- Find a most valuable subset of items with total weight W .
- Have to either take an item or not take it—can't take part of it.



2. Fractional Knapsack problem

- Like the 0-1 knapsack problem, but can take fraction of an item.

[2]. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. Page-430

1. Introduction to Greedy algorithm …(cont'd)

Example: knapsack problem

- Consider the following instance of the knapsack problem:

n=3, m=20,
(p1, p2, p3) = (25, 24, 15),
(w1, w2, w3) = (18, 15, 10)

- Fill the knapsack with the object with **largest profit**
- Assumption: If the object under consideration does not fit, then the fraction of it is included to fill the knapsack.

Solution

- Select Item-1 with profit p_1=25, here w_1=18, x_1=1.
- Remaining capacity = 20-18 = 2
- Select Item-2 with profit p_1=24, here w_2=15, x_1=2/15, profit (2/15 *24 =3.2)
- Remaining capacity = 0
- Total profit earned = 25+ 3.2= 28.2.

Therefore optimal solution is (x_1, x_2, x_3) = (1, 2/15, 0) with profit = 28.2

- xi is the fraction of ith object to be selected.

1. Introduction to Greedy algorithm …(cont'd)

Application of Greedy Method :-Job sequencing with deadline

- **Example:** Consider the following tasks with their deadlines and profits.

S. No.	1	2	3	4	5
Jobs	J1	J2	J3	J4	J5
Deadlines	2	2	1	3	4
Profits	20	60	40	100	80

Step 1

- Arrange the jobs in descending order of their profits.

S. No.	1	2	3	4	5
Jobs	J4	J5	J2	J3	J1
Deadlines	3	4	2	1	2
Profits	100	80	60	40	20

- The maximum deadline, d_m, is 4. Therefore, all the tasks must end before 4. **Choose the job with highest profit**.

- Since, the maximum deadline is met with **{J4, J3}** , the algorithm comes to an end. Thus output set of jobs scheduled within the deadline are {J4, J3}
- **Total Profit: 100 + 40 = 140**

[3]. Tutorials Point, Job Sequencing with Deadline. https://www.tutorialspoint.com/data_structures_algorithms/job_sequencing_with_deadline.htm

1. Introduction to Greedy algorithm …(cont'd)

Characteristics of a Greedy Algorithm

- The algorithm solves its problem by finding an optimal solution. This solution can be a maximum or minimum value.
- It makes choices based on the best option available.
- The algorithm is fast and efficient with time complexity of O(n log n) or O(n), therefore applied in solving large-scale problems.
- The search for optimal solution is done without repetition – the algorithm runs once.
- It is straightforward and easy to implement.

[4]. FeeCodeCamp, What is a Greedy Algorithm? Examples of Greedy Algorithms, antoluwa Heritage Alabi ,2023. https://www.freecodecamp.org/news/greedy-algorithms/

1. Introduction to Greedy algorithm …(cont'd)

How to Use Greedy Algorithms?

- Before applying a greedy algorithm to a problem, we need to ask two questions:
  - Do we need the best option at the moment from the problem?
  - Do we need an optimal solution (either minimum or maximum value)?
- If our answer to these questions is "Yes", then a greedy algorithm is a good choice to solve our problem.

[4]. FreeCodeCamp, What is a Greedy Algorithm?
Tantoluwa Heritage Alabi ,2023. https://www.freecodecamp.org/news/greedy-algorithms/

1. Introduction to Greedy algorithm …(cont'd)

Elements of the greedy strategy

- Determine the optimal substructure of the problem.
- Develop a recursive solution.
- Show that if we make the greedy choice
- Prove that it is always safe to make the greedy choice.
- Develop a recursive algorithm that implements the greedy strategy.
- Convert the recursive algorithm to an iterative algorithm.

[2]. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. Page-426

1. Introduction to Greedy algorithm …(cont'd)

Advantages and Drawback of Greedy Approach

- **Advantages of Greedy Approach**
  - Easier to describe.
  - Can perform better than other algorithms (but, not in all cases).

Drawback of Greedy Approach

- Doesn't always produce the optimal solution- the major disadvantage of the algorithm.

[5]. Programiz, Greedy Algorithm. https://www.programiz.com/dsa/greedy-algorithm

2. Type of Greedy Algorithms

Kruskal's Algorithm

- Kruskal’s algorithm finds a safe edge to add to the growing forest by finding of all the edges that connect any two trees in the forest, an edge (u, v) with the lowest weight.
- The algorithm first starts from the forest which is defined as a subgraph containing only vertices of the main graph of the graph, adding the least cost edges later until the minimum spanning tree is created **without forming cycles in the graph.**

- A minimum spanning tree is a sub graph that connects all the vertices present in the main graph with the least possible edges and minimum cost (sum of the weights assigned to each edge).

[2]. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. Page-592

2. Type of Greedy Algorithms …(cont'd)

Working of Kruskal's Algorithm

- Firstly, we will focus on sorting all the edges and will sort them according to their weight (low weight to high).
- Pick the edge having the least amount of weight and associate it with the spanning tree. If the edge being added forms a cycle, it should be rejected.
- Keep on adding the edges till all the vertices have been added and a minimum spanning tree has been formed.

[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/kruskal-algorithm-notes/

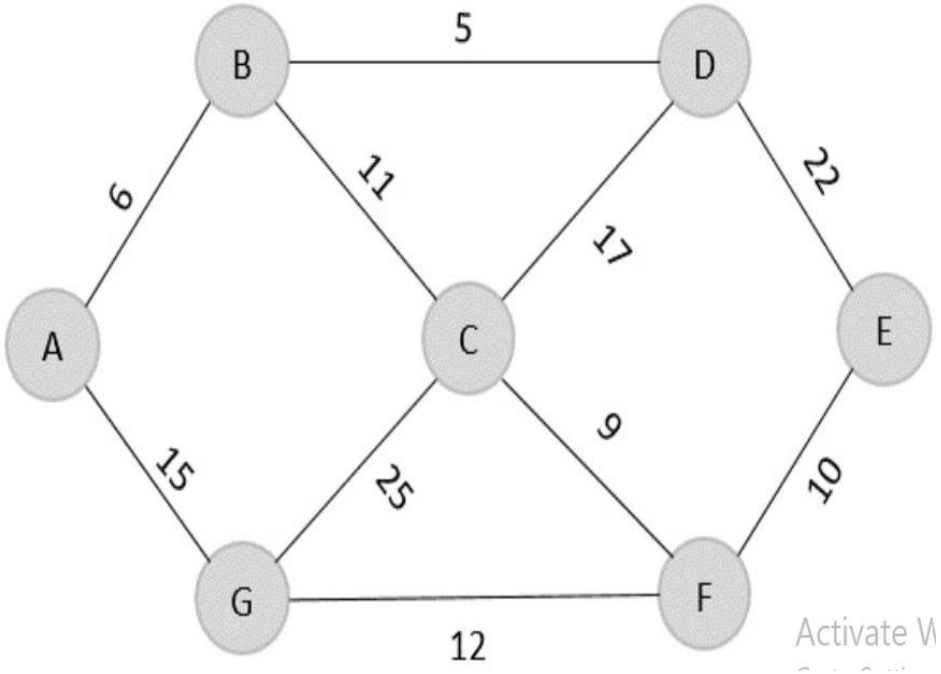
2. Type of Greedy Algorithms …(cont'd)

```
KRUSKAL(V, E):
    sort E by increasing weight
    F ← (V, ∅)
    for each vertex v ∈ V
        MAKESET(v)
    for i ← 1 to |E|
        uv ← ith lightest edge in E
        if FIND(u) ≠ FIND(v)
            UNION(u, v)
            add uv to F
    return F
```

- Initialize the Minimum Spanning Tree: Initially, it includes all vertices V and no edges
- For each vertex in V, a disjoint set is created allowing the algorithm to avoid cycles when adding edges
- A loop runs for each edge in the sorted list E. The uv denotes the ith lightest edge.
- Checks the endpoints of the edge uv belong to different sets. If they do, it means adding this edge will not form a cycle. Combining the sets of u and v
- If the vertices are in different sets, vertices are now connected.
- Finally, the function returns the set F, which now contains the edges of the Minimum Spanning Tree.

2. Type of Greedy Algorithms …(cont'd)

- **Examples**: Construct a minimum spanning tree using kruskals algorithm for the graph given below −

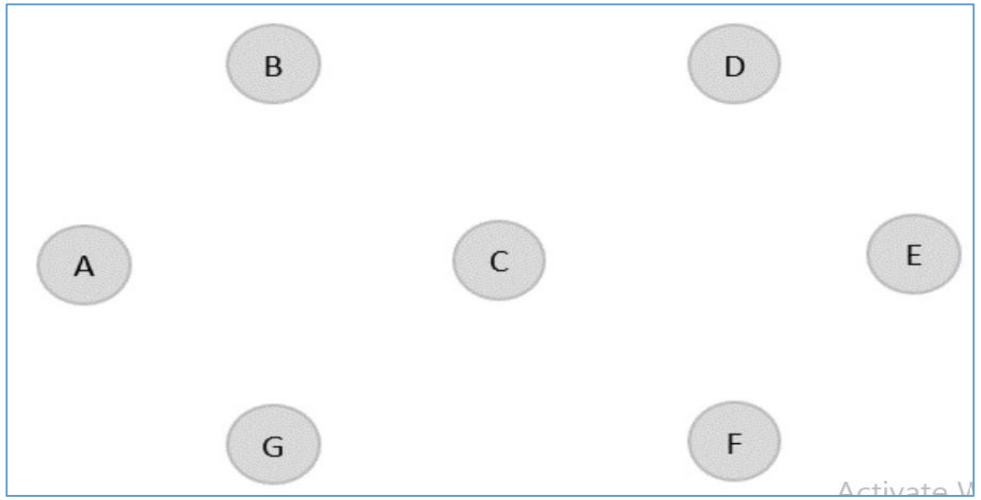


Step-1:- sort all the edges in the given graph in an ascending order and store the values in an array.

Edge	B→D	A→B	C→F	F→E	B→C	G→F	A→G	C→D	D→E	C→G
Cost	5	6	9	10	11	12	15	17	22	25

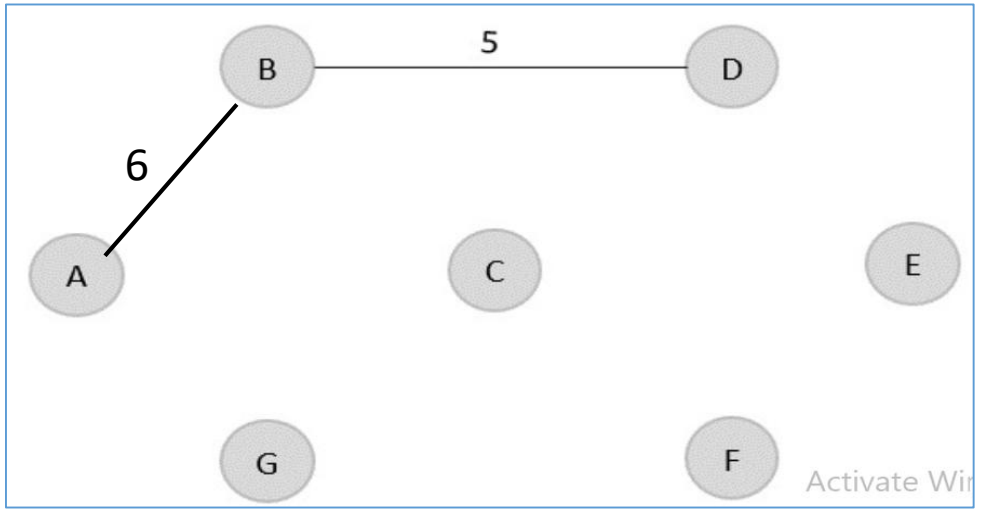
[3]. Tutorials Point Article , Kruskal's Spanning Tree Algorithm.
https://www.tutorialspoint.com/data_structures_algorithms/kruskals_spanning_tree_algorith

2. Type of Greedy Algorithms …(cont'd)



Step-2: Construct a forest of the given graph on a single plane.

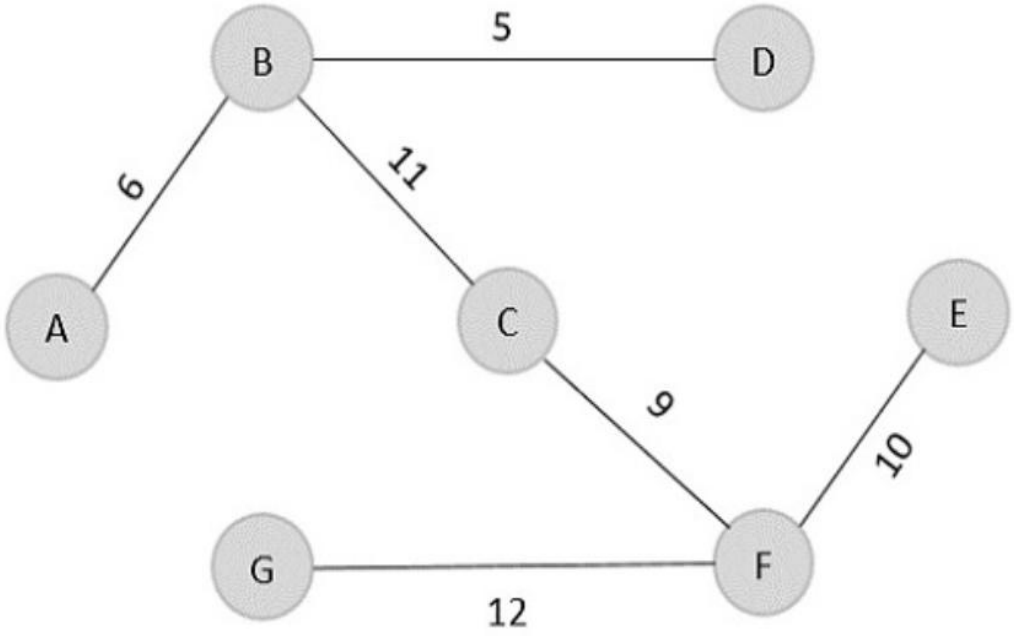
Step-3: From Step-1 (list of sorted edge costs), select the least cost edge and add it onto the forest in output graph.



Similarly, the next least cost edge is B → A = 6; so we add it onto the output graph.

[3].Tutorials Point Article , Kruskal's Spanning Tree Algorithm. https://www.tutorialspoint.com/data_structures_algorithms/kruskals_spanning_tree_algorithm

2. Type of Greedy Algorithms …(cont'd)



- The last edge from the least cost array to be added in the output graph is F → G = 12.

- Thus Finally, the obtained result is the minimum spanning tree of the given graph is with cost = 53.

[3]. TutorialsPoint, Kruskal's Spanning Tree Algorithm.
https://www.tutorialspoint.com/data_structures_algorithms/kruskals_spanning_tree_algorith

2. Type of Greedy Algorithms …(cont'd)

The Applications of Kruskal's Algorithm

- Can be used to build out electrical wiring across cities.
- It's possible to operate it to set up LAN connections.

The Complexity of Kruskal's Algorithm

- **Time Complexity**
  O(E logE) or O(V logV) is the time complexity of the Kruskal algorithm.
- Where E indicates the no. of edges, and V indicates the no. of vertices.

[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/kruskal-algorithm-notes/

2. Type of Greedy Algorithms …(cont'd)

Prim's Algorithm

- One of the efficient methods to find the minimum spanning tree of a graph.
- A **minimum spanning tree :** a sub graph that connects all the vertices present in the main graph with the least possible edges and minimum cost .
- The algorithm, similar to any shortest path algorithm, begins from a vertex that is set as a root and walks through all the vertices in the graph by determining the least cost **adjacent edges**.

[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/prims-algorithm-notes/

2. Type of Greedy Algorithms …(cont'd)

```
Algorithm Prim(G)
 Vt ← {v0}   // set of visited vertices
 Et ← Ø
 for i ← 1 to |V| - 1
 find minimum edge e between vertices v and u such that v is
   in Vt and u is in V-Vt
// add u to Vt
Vt ← Vt  U {u}
// add the edge to spanning tree
Et ← Et U {e}
```

This set keeps track of the vertices that have been included in the MST.

A set Et(age) is initialized to be empty.

Identifies the minimum-weight edge e that connects a vertex v from the visited set Vt to a vertex u that is not yet visited

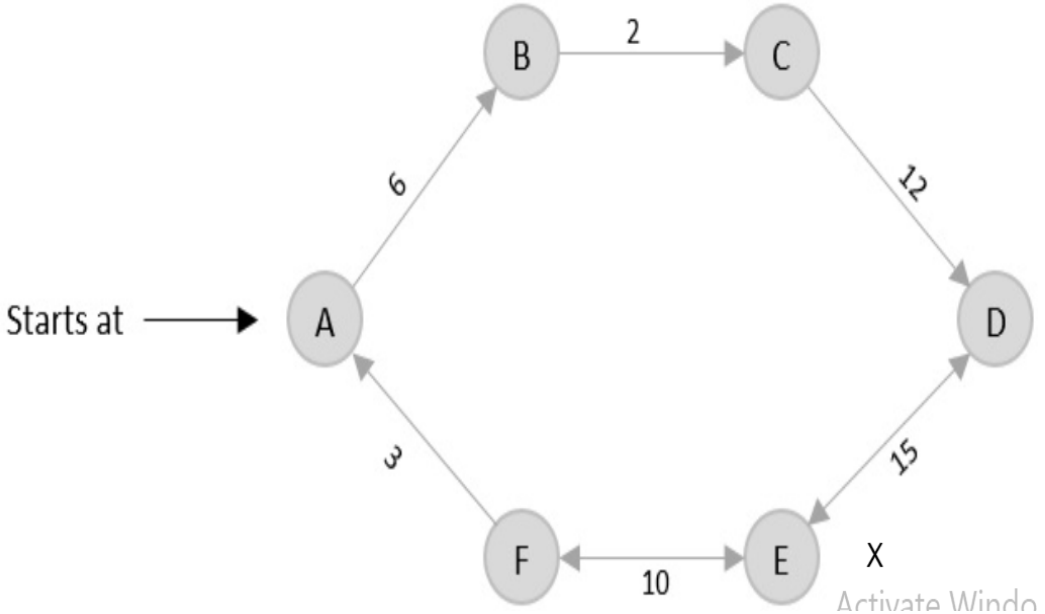
Finally, the function returns the set of edges Et

2. Type of Greedy Algorithms …(cont'd)

Prim's Algorithm

- To execute the prim's algorithm, the inputs taken by the algorithm are the graph G {V, E}, where V is the set of vertices and E is the set of edges, and the source vertex S.

- A minimum spanning tree of graph G is obtained as an output.



[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/prims-algorithm-notes/

2. Type of Greedy Algorithms …(cont'd)

Algorithm

- Stage 1: Choose a starting vertex
- Stage 2: Repetition of Steps 3 and 4 until the marginal vertex is present.
- Stage 3: Choose an edge with a minimum weight that connects the tree vertex and the fringe vertex.
- Stage 4: Include the chosen vertex and edge in the least spanning tree T.
- [FINISH OF LOOP]
- Stage 5: EXIT

[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/prims-algorithm-notes/

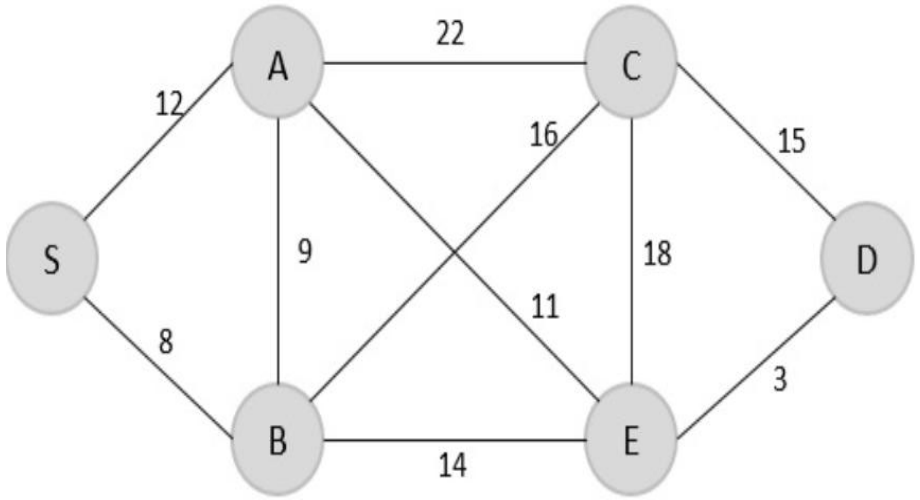
2. Type of Greedy Algorithms …(cont'd)

Examples

- Find the minimum spanning tree using prims method (greedy approach) for the graph given below with S as the arbitrary root.

Solution

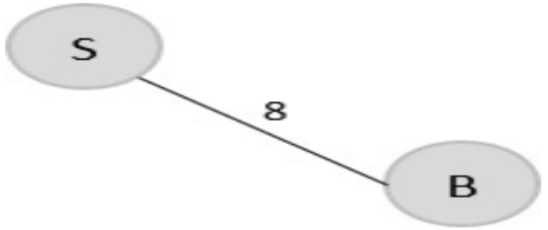
- **Step 1:** Create a visited array to store all the visited vertices into it.



- The arbitrary root is mentioned to be S, so among all the edges that are connected to **S** we need to find the least cost edge.

Step 2

Since B is the last visited, check for the least cost edge that is connected to the vertex B.

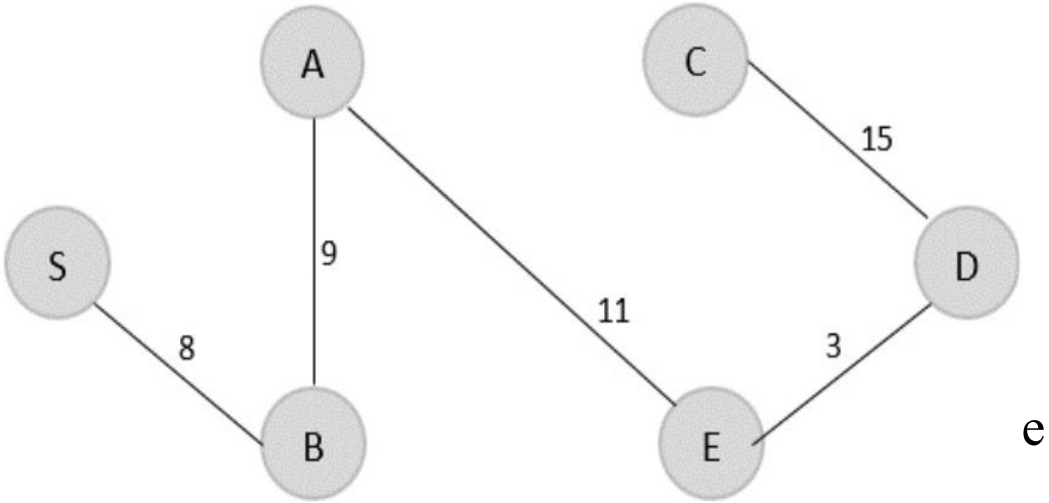


[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/prims-algorithm-notes/

2. Type of Greedy Algorithms …(cont'd)

Step 3

- Since A is the last visited, check for the least cost edge that is connected to the vertex A.



e.

V = {S, B, A, E, D, C}

- The minimum spanning tree is obtained with the minimum cost = 46

- But A → B is already in the spanning tree, check for the next least cost edge. Hence, A → E is added to the spanning tree.

Step 4

- Since E is the last visited, check for the least cost edge that is connected to the vertex E.

Step 5

- Since D is the last visited, check for the least cost edge that is connected to the vertex D.

2. Type of Greedy Algorithms …(cont'd)

Time Complexity

- Each edge insertion and extraction from the priority queue contributes **O(log V)**
  - Thus, processing all edges takes **O(E log V)** **t**ime in total.

Prim's Algorithm Application

- Installing electrical wiring lines
- In designing the network
- To build protocols in network cycles

[6]. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/prims-algorithm-notes/

2. Type of Greedy Algorithms …(cont'd)

Huffman Coding

- A technique of compressing data to reduce its size without losing any of the details.
- It was first developed by David Huffman.
- Useful to compress the data in which there are frequently occurring characters.

[8]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)

How Huffman Coding works?

- Suppose the string below is to be sent over a network.

- Each character occupies 8 bits.
- There are a total of 15 characters in the above string.
- Thus, a total of **8 * 15 = 120** bits are required to send this string.

- Using the Huffman Coding technique, we can compress the string to a smaller size.
- Huffman coding first creates a tree using the frequencies of the character and then generates code for each character.

2. Type of Greedy Algorithms …(cont'd)

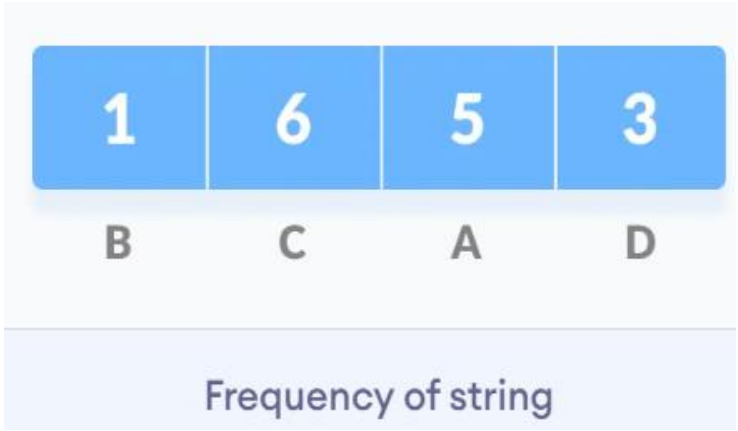
- Once the data is encoded, it has to be decoded. Decoding is done using the same tree.
- Huffman Coding prevents any ambiguity in the decoding process using the concept of **prefix code** i.e. a code associated with a character should not be present in the prefix of any other code.
- The tree created above helps in maintaining the property.

[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)

- Huffman coding is done with the help of the following steps.

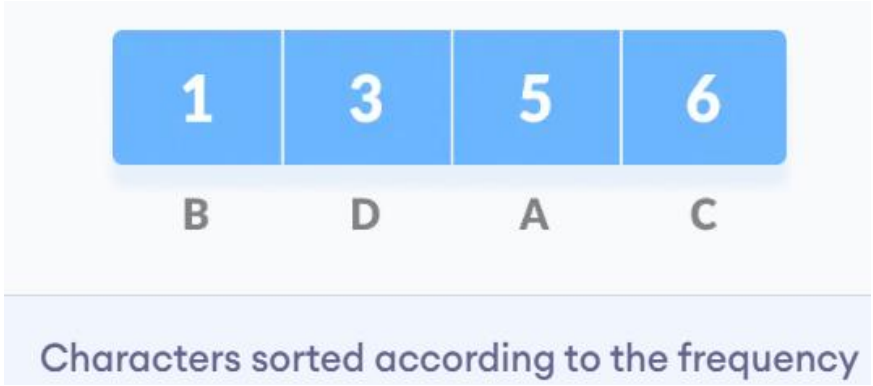
Step-1



Step-1 Calculate the frequency of each character in the string

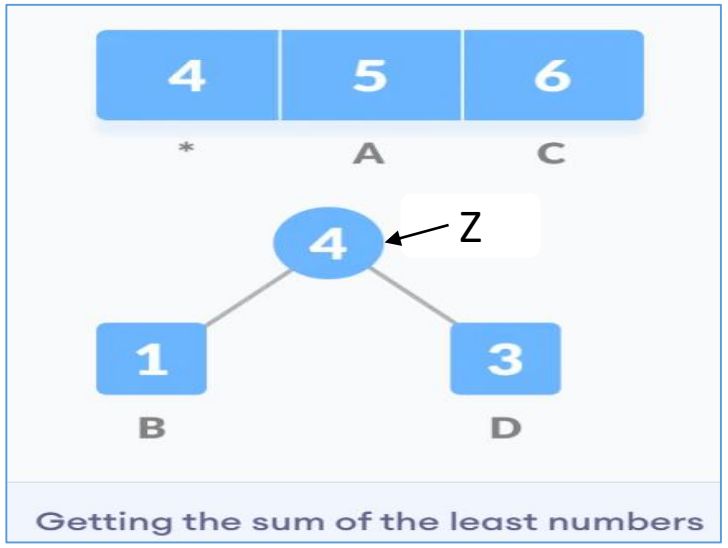
Step2- Sort the characters in increasing order of the frequency. These are stored in a priority queue Q.

Step-2

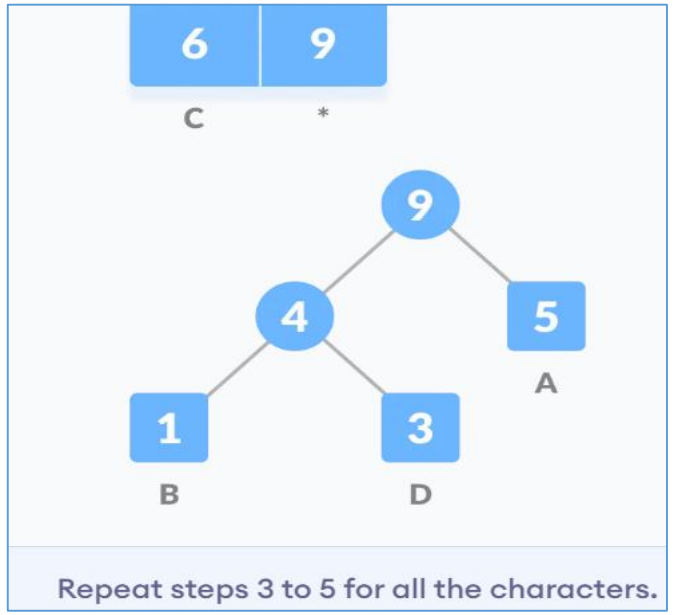


[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)



Getting the sum of the least numbers

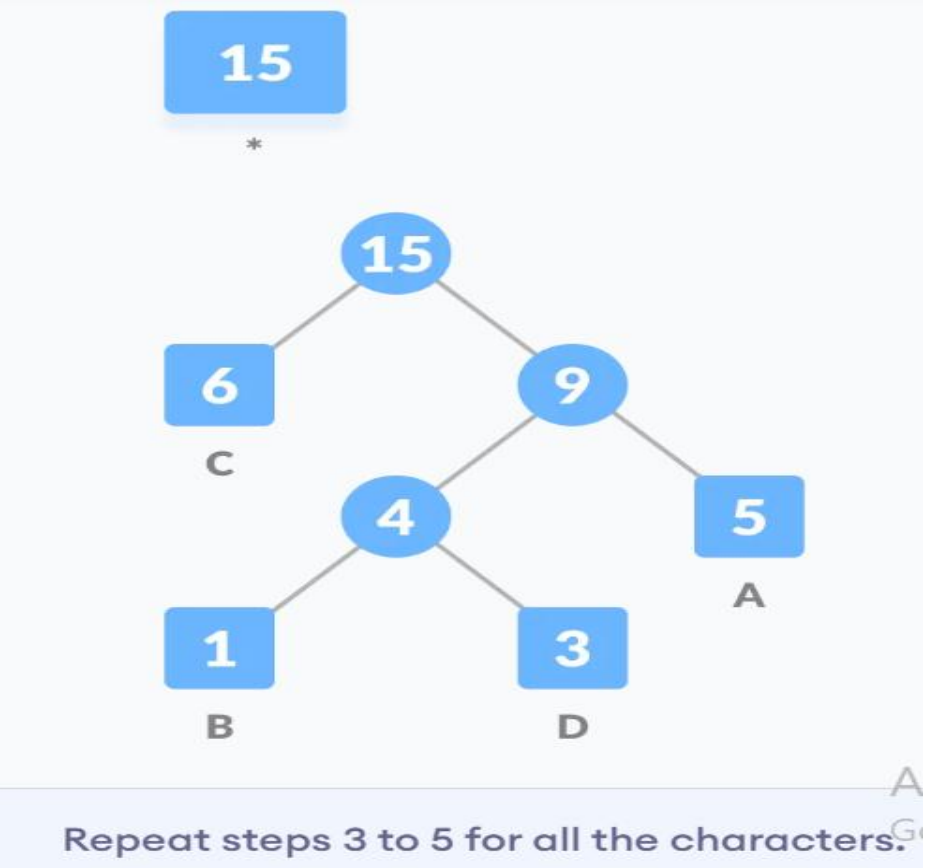


Repeat steps 3 to 5 for all the characters.

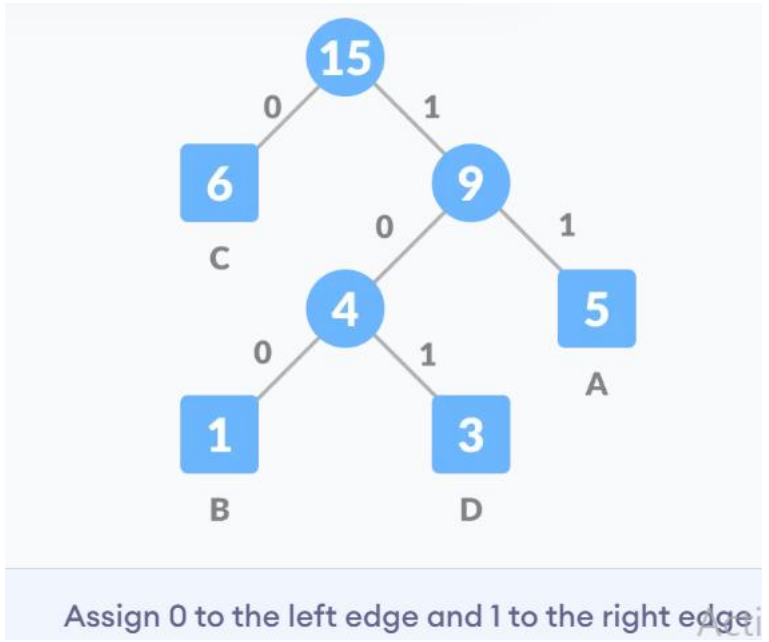
Step-3. Make each unique character as a leaf node.
Step-4 .Create an empty node z. Assign the minimum frequency to the left child of z and assign the second minimum frequency to the right child of z. Set the value of the z as the sum of the above two **minimum frequencies**.

Step-5 Remove these two minimum frequencies from Q and add the sum into the list of frequencies
Step-6 Insert node z into the tree.
Step-7 Repeat steps 3 to 5 for all the characters.

2. Type of Greedy Algorithms …(cont'd)



Step-8: For each non-leaf node, assign 0 to the left edge and 1 to the right edge.

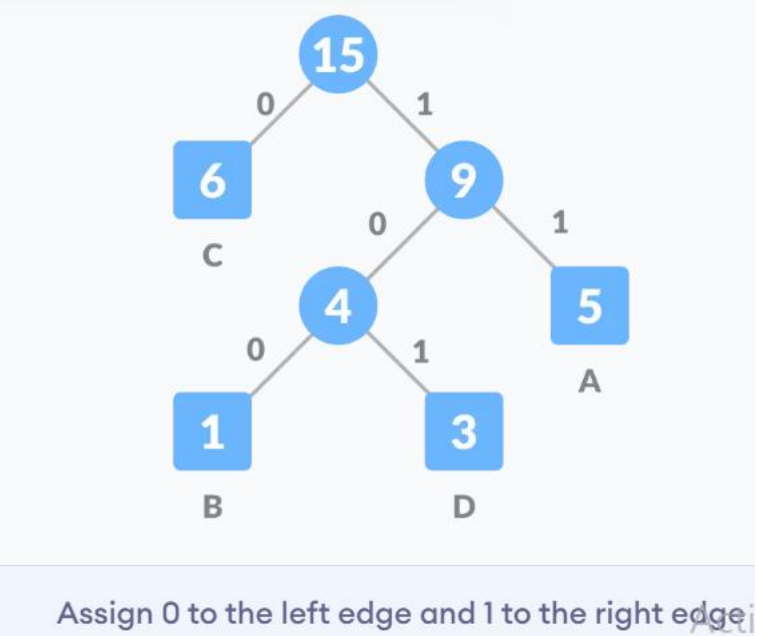


[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)

Greedy Algorithm for Huffman Code

- According to Huffman algorithm, a bottom up tree is built starting from the leaves. Initially, there are **n** singleton trees in the forest, as each tree is a leaf.
- **Greedy strategy**:- first finds two trees having minimum frequency of occurrences. Then these two trees are merged in a single tree where the frequency of this tree is the total sum of two merged trees.
- The whole process is repeated until there in only one tree in the forest.



[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)

- For sending the above string over a network, we have to send the tree as well as the above compressed-code.
- The total size is given by the table below.

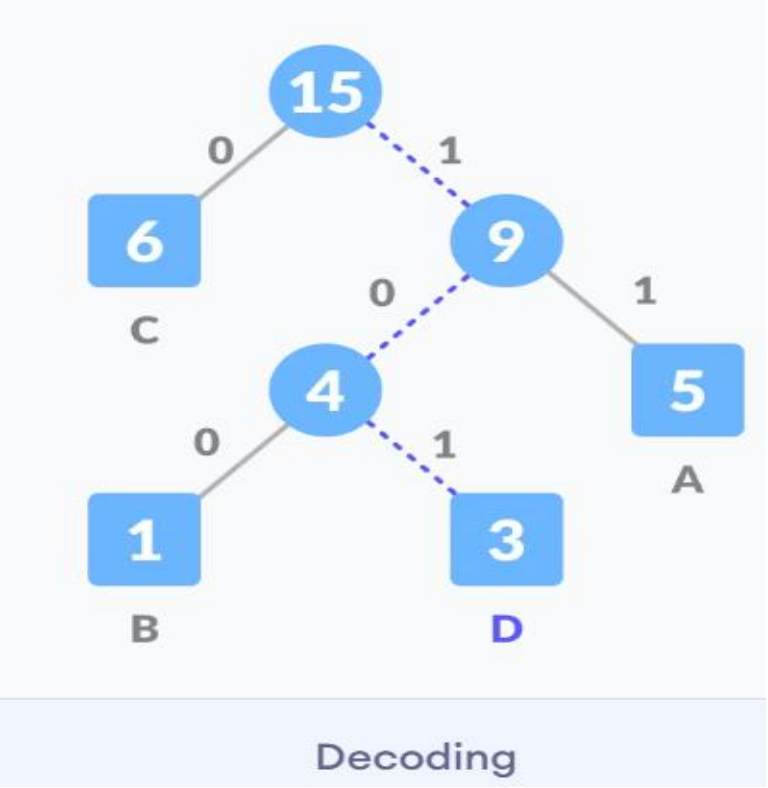
Character	Frequency	Code	Size
A	5	11	5*2 = 10
B	1	100	1*3 = 3
C	6	0	6*1 = 6
D	3	101	3*3 = 9
4 * 8 = 32 bits	15 bits		28 bits

- **Without encoding**, the total size of the string was 120 bits. After encoding the size is reduced to **32 + 15 + 28 = 75**.

2. Type of Greedy Algorithms …(cont'd)

Decoding the code

- For decoding the code, we can take the code and traverse through the tree to find the character.
- Let 101 is to be decoded, we can traverse from the root as in the figure below.



Decoding

[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

2. Type of Greedy Algorithms …(cont'd)

Huffman Coding Complexity

- The time complexity for encoding each unique character based on its frequency is O(nlog n).
- Thus the overall complexity is O(n log n).

Huffman Coding Applications

- Huffman coding is used in conventional compression formats like GZIP, BZIP2, PKZIP, etc.
- For text and fax transmissions.

[7]. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

Summary

- Greedy Method is the most straight forward design technique, used to determine a feasible solution that may or may not be optimal.
- The algorithm is fast and efficient with time complexity of O(n log n) or O(n), therefore applied in solving large-scale problems.
- To apply a greedy algorithm to a problem, need to ask the best option at the moment from the problem and need of an optimal solution (either minimum or maximum value).
- Advantages of Greedy Approach: The algorithm is easier to describe and can perform better than other algorithms (but, not in all cases).
- Drawback of Greedy Approach: it doesn't always produce the optimal solution.
- While Prim's and Kruskal's algorithms tackle graph problems and Huffman coding addresses data compression, they share the fundamental greedy strategy: making local optimal choices to derive a globally optimal solution.

References

1. S.K. Sathua , M.R. Kabat, R. Mohanty. Design And Analysis Of Algorithms., https://vssut.ac.in/lecture_notes/lecture1428551222.pdf

2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 2009. Page-430.

3. Tutorials Point, Job Sequencing with Deadline. https://www.tutorialspoint.com/data_structures_algorithms/job_sequencing_with_deadline.htm

4. FreeCodeCamp, What is a Greedy Algorithm? Examples of Greedy Algorithms,

Tantoluwa Heritage ,2023. https://www.freecodecamp.org/news/greedy-algorithms/

5. Programiz, Greedy Algorithm. https://www.programiz.com/dsa/greedy-algorithm

6. BYJU'S, Kruskal Algorithm - GATE CSE Notes. https://byjus.com/gate/kruskal-algorithm-notes/

7. Programiz, Huffman Coding Algorithm. https://www.programiz.com/dsa/huffman-coding

Thank You!

For your attention