Greedy Method

Outline

- Introduction
- Optimization Problem
- Characteristics And Features
- Greedy Properties.
- Greedy Approach
  - The Coin Changing Problem.
  - Traveling Salesman.
- MST
- Kruskal's And Prim's Algorithm.
- Huffman Code.
- Advantages And Disadvantages

Greedy Introduction

- Greedy algorithms are simple and straightforward.
- They are shortsighted in their approach
- A greedy algorithm is similar to a dynamic programming algorithm, but the difference is that solutions to the sub problems do not have to be known at each stage
- It is used to solve the optimization problems.

Optimization Problems

- An optimization problem is one in which you want to find, not just a solution, but the best solution
- A "greedy algorithm" sometimes works well for optimization problems
- A greedy algorithm works in phases. At each phase:
  - You take the best you can get right now, without regard for future consequences
  - You hope that by choosing a local optimum at each step, you will end up at a global optimum

Characteristics And Features

- To construct the solution in an optimal way. Algorithm Maintains two sets
  - One contains chosen items and
  - The other contains rejected items.
- Greedy algorithms make good local choices in the hope that They result in
  - An optimal solution.
  - Feasible solutions.

Greedy Property

It consists of two property.

1. **Greedy-Choice Property:**

   It says that a globally optimal solution can be arrived at by making a locally optimal choice.

2. **Optimal Substructure:**

   An optimal global solution contains the optimal solutions of all its sub problems.

Greedy Approach

- Greedy Algorithm works by making the decision that seems most promising at any moment; it never reconsiders this decision, whatever situation may arise later.
- As an example consider the problem of "Making Change".
- Coins available are:
  - ✓ 100 cents
  - ✓ 25 cents
  - ✓ 10 cents
  - ✓ 15 cents
  - ✓ 1 cent

CONTINUED…

Problem:

Make a change of a given amount using the smallest possible number of coins.

Solution:

- The coin is selected using greedy criterion: at each stage increases the total amount of change constructed as much as possible.
- To assure feasibility i.e. the change given exactly equals the desired amount of the solution

Coin Changing Problem

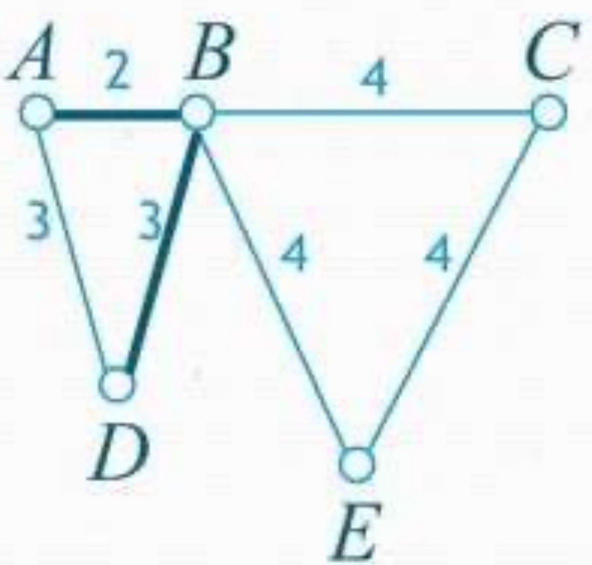
Algorithm:

Make change for n units using the least possible number of coins.

```
MAKE-CHANGE (n)
   C ← {100, 25, 10, 15, 1}    // constant.
   S← {};                      // set that will hold the solution set.
    Sum ← 0 sum of item in solution set
    WHILE sum not = n
       x = largest item in set C such that sum + x ≤ n
       IF no such item THEN
           RETURN    "No Solution"
       S ← S {value of x} & Remove x from C.
       sum ← sum + x
    RETURN S
```

Traveling Salesman

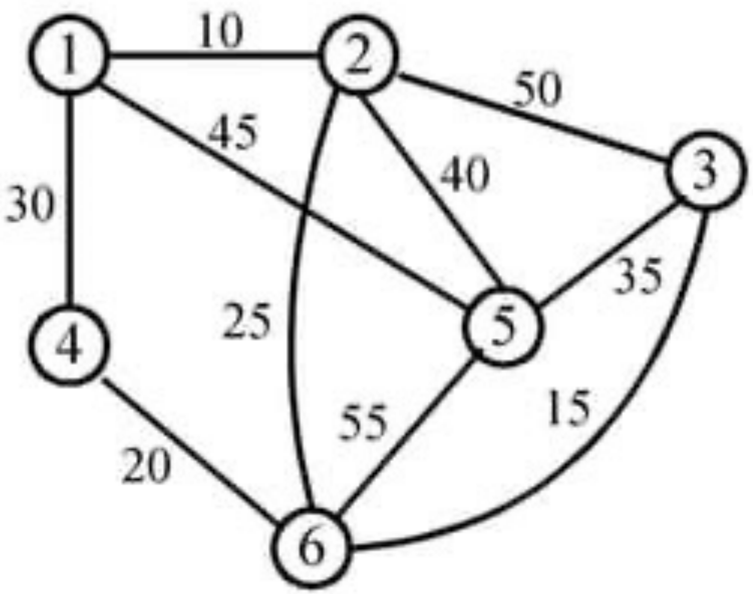
- A salesman must visit every city (starting from city *A*), and wants to cover the least possible distance
- He can revisit a city (and reuse a road) if necessary
- He does this by using a greedy algorithm: He goes to the next nearest city from wherever he is

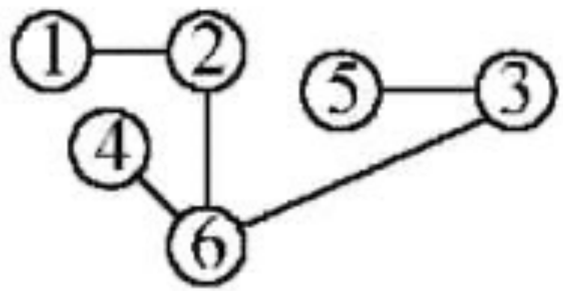


- ✓ From *A* he goes to *B*
- ✓ From *B* he goes to *D*
- ✓ This is *not* going to result in a shortest path!
- ✓ The best result he can get now will be ***ABDBCE***, at a cost of **16**
- ✓ An actual least-cost path from *A* is ***ADBCE***, at a cost of **14**

An Example Of MST

- A graph and one of its minimum costs spanning tree.
- Application of MST.





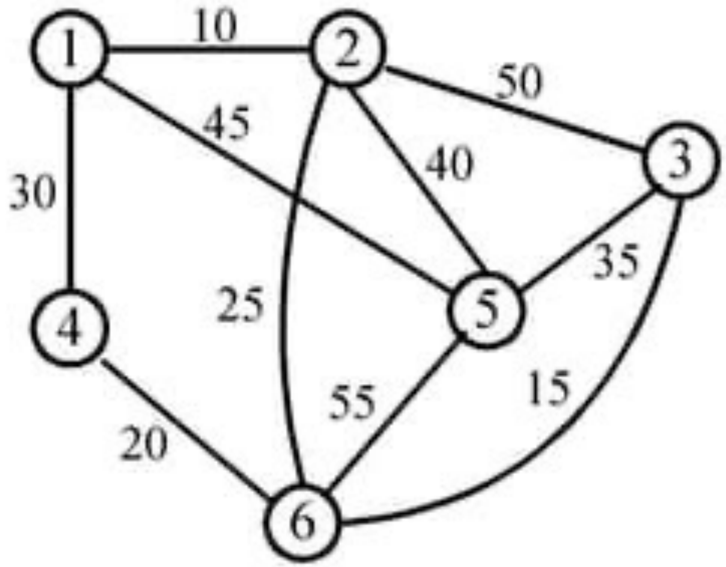
Kruskal's Algorithm For Finding MST

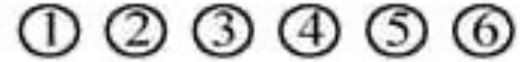
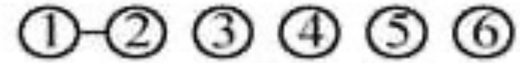
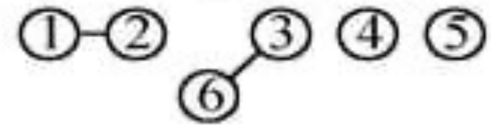
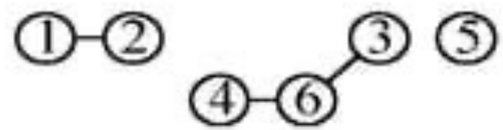
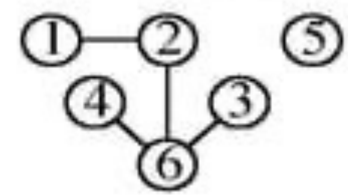
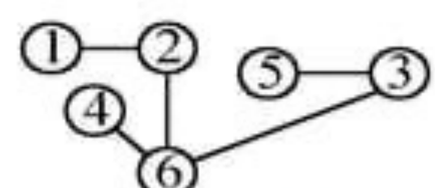
Step 1: Sort all edges into non decreasing order.

Step 2: Add the next smallest weight edge to the forest if it will not cause a cycle.

Step 3: Stop if n-1 edges. Otherwise, go to Step2.

An Example Of Kruskal's Algorithm

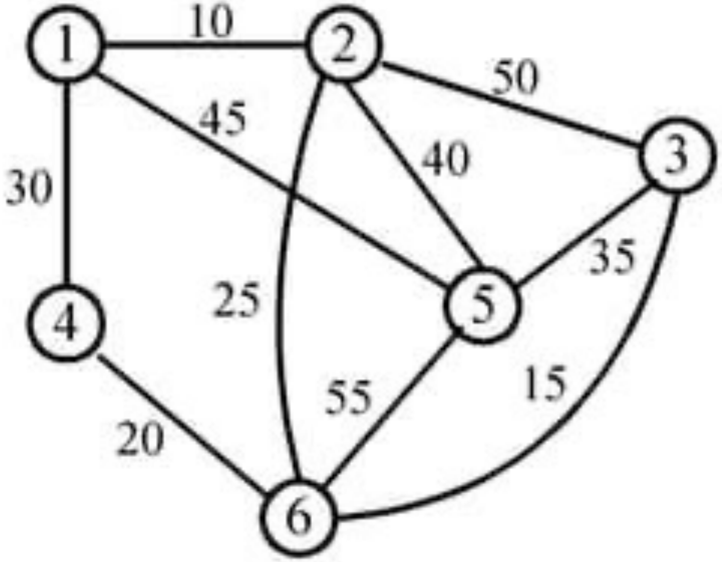


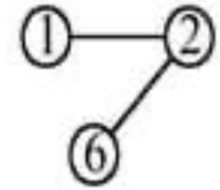
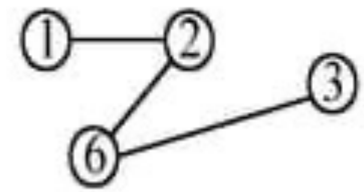
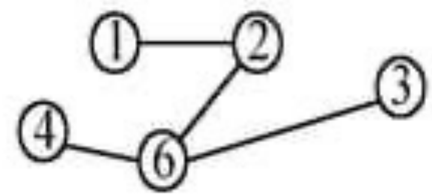
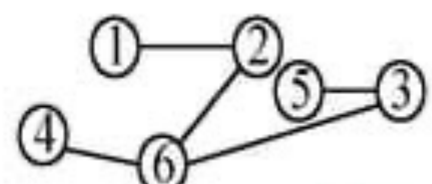
Edge	Cost	Spanning Forest
		
(1,2)	10	
(3,6)	15	
(4,6)	20	
(2,6)	25	
(1,4)	30	(reject)
(3,5)	35	

Prim's Algorithm For Finding MST

- Initialize a tree with a single vertex, chosen arbitrarily from the graph.
- Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
- Repeat step 2 (until all vertices are in the tree).And E=V-1.

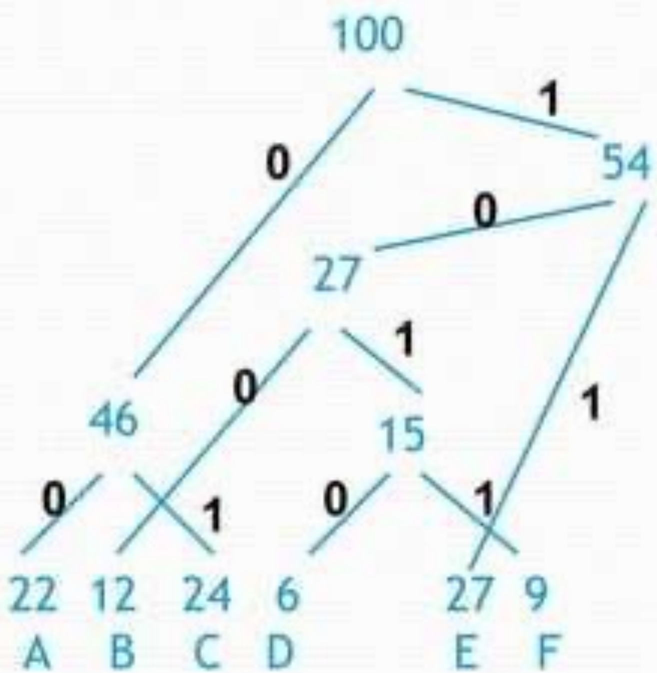
An Example For Prim's Algorithm



Edge	Cost	Spanning tree
(1,2)	10	
(2,6)	25	
(3,6)	15	
(6,4)	20	
(3,5)	35	

Huffman Encoding

- The Huffman encoding algorithm is a greedy algorithm
- You always pick the two smallest numbers to combine



A=00
B=100
C=01
D=1010
E=11
F=1011

- The Huffman algorithm finds an optimal solution

Advantages And Disadvantages

Advantages:

- They are easier to implement.
- They require much less computing resources.
- They are much faster to execute.
- Greedy algorithms are used to solve optimization problems

Disadvantages:

- Their only disadvantage being that they not always reach the global optimum solution.
- On the other hand, even when the global optimum solution is not reached, most of the times the reached sub-optimal solution is a very good solution.


Thank You!


Have A Nice Day!
©micka