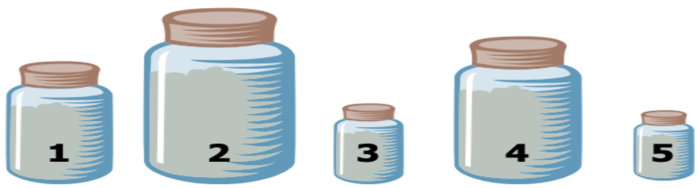
Lesson-16- Shortest Path

- **An optimal solution** is a feasible solution that results in the largest possible objective function value when maximizing (or smallest when minimizing).
- **The feasible solution:** A subset of given inputs that satisfies all specified constraints of a problem is known as a "feasible solution".

Slide-11 - Fractional Knapsack Problem

Given a set of 5 items, find the maximum total profit with the at most weight of **10** ml.
W = 10 ml

Objects/Items



Benefits/Profits In Dollar($) Bi	12	32	40	30	50
Weights in ml(Wi)	4	8	2	6	1
Value $ per ml Bi/Wi	3	4	20	5	50
Solution set Xi (included or not)	0	1/8	1	1	1

Select the Bi/Wi Highest to lowest

Step 1: Pick 50 → Item 5 with the weight 1. Include in the set Xi and deduct its weight from the W. 10 – 1 = 9

Step 2: Pick 20 → Item 3 with the weight 2. Include in the set Xi and deduct its weight from the remining weight W. 9 – 2 = 7

Step 3: Pick 5 → Item 4 with the weight 6. Include in the set Xi and deduct its weight from the remining weight W. 7 – 6 = 1

Step 4: Pick 4 → Item 2 with the weight 1 from the 8(Select the fraction). Include in the set Xi and deduct its weight from the remining weight W. 7 – 6 = 1

Let's verify the weight we included is at most W = 10

Sum of Xi * Wi = 0 * 4 + (1/8) * 8 + 1 * 2 + 1 * 6 + 1 * 1

= 0 + 1 + 2 + 6 + 1 = 10

Let's calculate total Profit
Sum of Xi * Bi = 0 * 12 + (1/8)*32 + 1 * 40 + 1 * 30 + 1 * 50

= 0 + 4 + 40 + 30 + 50
= $ 124

<u>Slide-12</u>

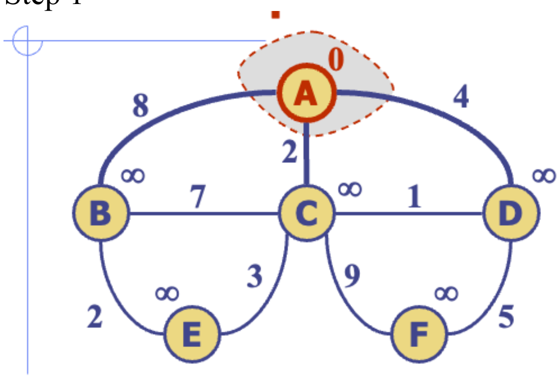
Run time: O(n log n). Why?

Answer: This is typically because the items must first be sorted by their value-to-weight ratio before the greedy approach is applied, and sorting can be done in O(n log n) time with comparison-based sorting algorithms like quicksort or mergesort.

Slide-35 – Last Point Details

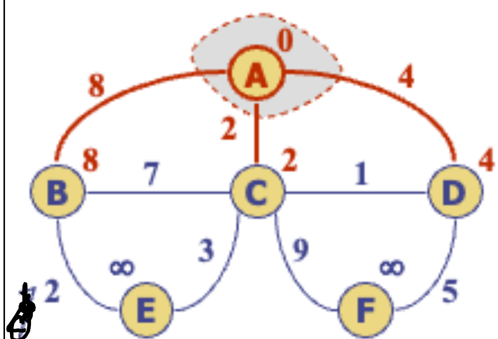
Correctness: The argument for correctness is that because edge weights are non-negative and the algorithm relaxes edges (updating distances if a shorter path is found), the distance to a node once it is added to the tree cannot be incorrect. Therefore, as long as the distance from the source to F (d(F)) is greater than or equal to the distance from the source to D (d(D)), the algorithm has not made an error in the distance for vertex F.

Dijkstra’s Algorithm

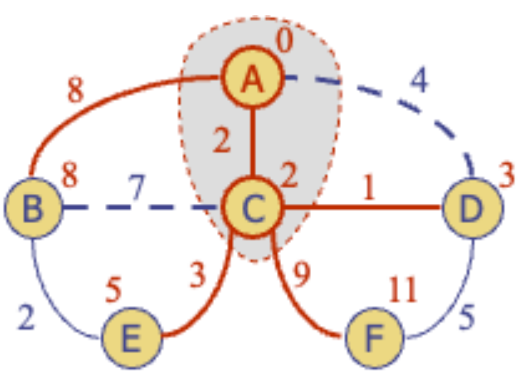
<table>
<tr>
<td>
Start Graph
Step 1


</td>
<td>
Formula
$d(z) \leftarrow \min\{d(z), d(u)+weight(e)\}$

Initial Vertex A
d(A) = 0
d(B) = ∞
d(C) = ∞
d(D) = ∞
d(E) = ∞
d(F) = ∞

A is in the Tree
</td>
</tr>
<tr>
<td>
Step 2


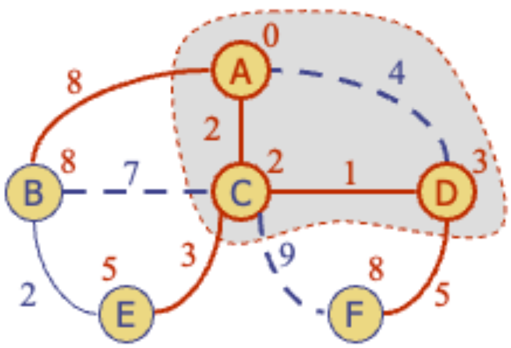
</td>
<td>
Adjacent to A are B, C, D
You have to pick minimum distance

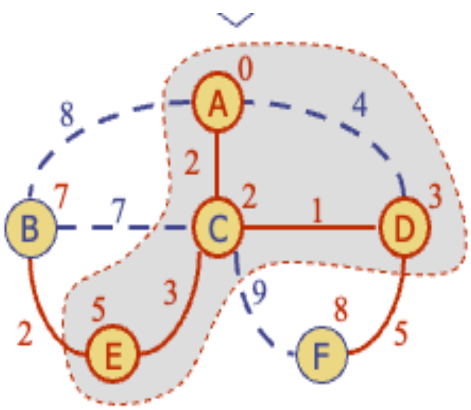
d(B) = min{∞, d(u) + weight(b)}
= min(∞, 0 + 8}
= 8
d(C) = min{∞, 0 + 2 } = 2 (min)
d(D) = min(∞, 0 +4 } = 4
d(A) = 0
d(B) = 8
d(C) = 2
d(D) = 4
d(E) = ∞
d(F) = ∞
</td>
</tr>
<tr>
<td>
Step 3, Add C in the tree


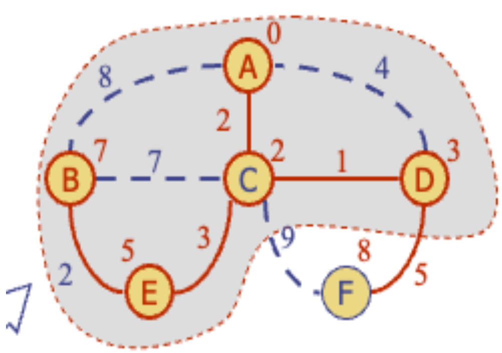
</td>
<td>
Pick the edges adjancent A & C

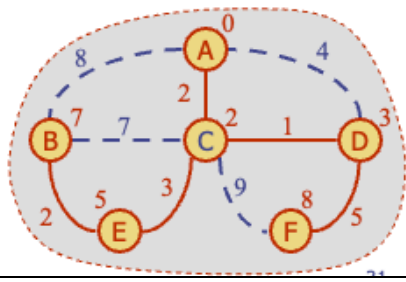
D(c) = min {4, d(c) + weight(d)}
Min(4, 2 + 1) = 3
D(e) = min { ∞ , 2 + 3} = 5

d(A) = 0
d(B) = 8
d(C) = 2
d(D) = 3
d(E) = 5 (min)
</td>
</tr>
</table>

<table>
<tr><td></td><td>d(F) = ∞</td></tr>
<tr><td>Step 4, Add D is in the tree
</td><td>Update the distances from A, C, D

D(f) = { ∞, 3 + 5 } = 8
d(A) = 0
d(B) = 8
d(C) = 2
d(D) = 3
d(E) = 5
d(F) = 8</td></tr>
<tr><td>Step 5, Add E in the cloud
</td><td>$d(z) \leftarrow \min\{d(z), d(u)+weight(e)\}$

d(A) = 0
d(B) = 7
d(C) = 2
d(D) = 3
d(E) = 5
d(F) = 8
d(b) = { 8, 5 + 2} = 7</td></tr>
<tr><td>Step 6
</td><td>d(A) = 0 -->
d(B) = 7 – A – C – E - B
d(C) = 2 A - C
d(D) = 3 A – C – D
d(E) = 5. A – C - E
d(F) = 8. A – C - D - F</td></tr>
</table>

Step 7 A B C D E F 0 8 4 2 7 7 2 1 3 5 3 9 8 2 5	