Chapter 11

11.1 Let the jobs be J_1=4, J_2=5 and J_3=10 Consider a scheduling problem without deadlines and find the optimal scheduling order of the jobs.

Solution:

Schedule	Total time in System	Average Time
1,2,3	4+(4+5)+(4+5+10)	32/3 = 10.6
1,3,2	4+(4+10)+(4+10+5)	27/3 = 12.3
2,1,3	5+(5+4)+(5+4+10)	33/3 = 11.0
2,3,1	5+(5+10)+(5+10+4)	39/3 = 13.0
3,1,2	10+(10+4)+(10+4+5)	43/3 = 14.3
3,2,1	10+(10+5)+(10+5+4)	44/3 = 14.6

It can be seen that the best policy is shortest job first i.e., 1,2,3

11.2 Let the jobs be J_1=8, J_2=9 and J_3=9. Consider a scheduling problem without deadlines and find the optimal scheduling order of the jobs.

Solution:

Using the above principle and model, the best scheduling policy is to assign shortest jobs first. ∴ the best order is 8,9,9.

11.3 Apply the greedy approach to a machine scheduling problems where deadlines are involved. One can have profit only if the task is completed within the deadlines else, even if the task is completed, it is useless. The details of the jobs are as follows:

Job	Deadline	Profit

1	1	20
2	1	35
3	2	15
4	1	40

Solution:

The schedule order would be

		Profit
{1}	20	
{2}	35	
{3}	15	
{4}	40	
{1,2}	Not feasible	–
{1,3}	✓	35
{1,4}	Not feasible	–
{2,1}	Not feasible	–
{2,3}	✓	55
{2,4}	Not feasible	–
{3,1}	Not feasible	–
{3,2}	Not feasible	–
{3,4}	Not feasible	–
{4,1}	Not feasible	–
{4,2}	Not feasible	–
{4,3}	✓	55

All higher order sequences are not possible.

∴ The maximum profit is 55.

11.4 Apply the greedy approach to machine scheduling problem where deadlines involved. The details of the jobs are as follows:

Job	**Deadline**	**Profit**
1	1	10

2	3	75
3	2	15
4	4	40

Find optimal scheduling order of the jobs. Consider the deadline constraints.

Solution:

The Possible sequences are listed below:

	Profit
{1}	10
{2}	75
{3}	15
{4}	40
{1,2}	85
{1,3}	35
{1,4}	50
{2,1}	Not feasible
{2,3}	90
{2,4}	75 + 40 = 115
{3,1}	Not feasible
{3,2}	75 + 15 = 90
{3,4}	15 + 40 = 55
{4,1}	Not feasible
{4,2}	40 + 75 = 115
{4,3}	40 + 15 = 55
⋮	

So one can check that all the jobs can be scheduled as { 1, 3, 2, 4 } without affecting deadline.

∴ The total profit is 10 + 15 + 75 + 40 = 140.

11.5 Apply the greedy approach to an activity selection problem, along with the starting and finish times of the jobs, which are given in the table. Find the optimal order.

Solution:

Job	Start Time	Finish Time
1	1	3
2	2	4
3	4	6
4	5	8
5	7	9

Assuming, picking up the job first 1, then the next selected sequence would be

$1 \to 3 \to 5$

Considering the criteria that Job3 start time is greater than finish time of Job1. Similarly Job5 is also selected.

11.6 Use an activity selection algorithm to schedule the following activities:

Job	Start Time	Finish Time
1	0	4
2	3	6
3	5	7
4	6	9
5	10	12
6	13	16
7	15	19

Find optimal order?

Solution:

Assuming Job1 is selected, the selected sequence would be

$1 \to 3 \to 5 \to 6.$

11.7 A Knapsack capacity is 100. The weights and values of five objects are as follows:

Weight w_i :	10	20	30	40	50

Value v_i **:**	20	30	66	20	60

Solve the Knapsack problem and find the maximum profit that can be obtained.

Solution:

$$\frac{v_1}{w_1} = \frac{20}{10} = 2$$

$$\frac{v_2}{w_2} = \frac{30}{20} = 1.5$$

$$\frac{v_3}{w_3} = \frac{66}{30} = 2.2$$

$$\frac{v_4}{w_4} = \frac{20}{40} = 0.5$$

$$\frac{v_5}{w_5} = \frac{60}{50} = 1.2$$

Sort the items in sorted order

$$\begin{array}{ccccc} \mathit{Item}3 & 1 & 2 & 5 & 4 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \{2.2, & 2, & 1.5, & 1.2, & 0.5\} \end{array}$$

Fill item 3 remove weight {100-30} = 70

Fill item 1 remove weight {70-10} = 60

Fill item 2 remove weight {60-20} = 40

Item 5 remove weight $= \frac{40}{50} = 0.8$

∴The order is {1, 1, 1, 0, 0.8 }

∴ Profit = 20+30+66+0.8×60 = 164.

11.8 A Knapsack can carry weights not exceeding 100. The weights and values of gfive objects are as follows:

Weight w_i **:**	10	20	30	40	50
Value p_i **:**	23	34	66	20	10

Solve the Knapsack problem and find the maximum profit that can be obtained.

Solution:

$$\frac{p_1}{w_1} = \frac{23}{10} = 2.3$$

$$\frac{p_2}{w_2} = \frac{34}{20} = 1.7$$

$$\frac{p_3}{w_3} = \frac{66}{30} = 2.2$$

$$\frac{p_4}{w_4} = \frac{20}{40} = 0.5$$

$$\frac{p_5}{w_5} = \frac{10}{50} = 0.2$$

∴ The sorted item are { 2.3, 2.2, 1.7, 0.5, 0.2 }

Fill Item 1 remove weight { 100 – 10 } = 90

Item 3 remove weight { 90 – 30 } = 60

Item 2 remove weight { 60 – 20 } = 40

Item 4 remove weight { 40 – 40 } = 0

Item 5 Can't be filled

∴ { 1, 1, 1, 1 }

∴ Profit = 143

11.9 Lengths of the program of jobs J_1, J_2 and J_3 are as follows : $J_1 = 4$, $J_2 = 5$, $J_3 = 10$. What is the optimal order for storing these programs?

Solution:

Schedule	Total time in System	Average Time
1,2,3	4+(4+5)+(4 +5+10)	32/3 = 10.67
1,3,2	4+(4+10)+(4+10+5)	37/3 = 12.33
2,1,3	5+(5+4)+(5+4+10)	33/3 = 11

2,3,1	5+(5+10)+(5+10+4)	39/3 = 13
3,1,2	10+(10+4)+(10+4+5)	43/3 = 14.33
3,2,1	10+(10+5)+(10+5+4)	44/3 = 14.67

∴ The optimal order is { 1, 2, 3 }.

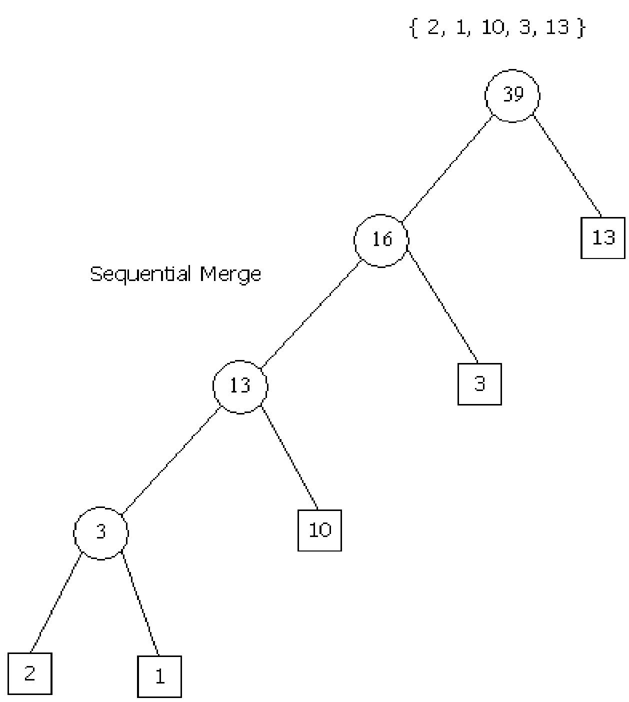
11.10 Lengths of the programs of jobs J_1, J_2, J_3 and J_4 are as follows: $J_1 = 4$, $J_2 = 5$, $J_3 = 10$ and $J_4 = 3$. What is the optimal order for storing the programs?

Solution:

Using the shortest job first order principle, one can conclude that the optimal order is {4, 1, 2, 3 }.

11.11 Length of the programs of Jobs J_1,J_2,J_3, J_4 and J_5 are as follows: J_1 =2,J_2=1,J_3 =10, J_4 = 3 and J_5 = 13. What is the optimal merge order using greedy approach? Compare that with the sequential merge approach? Compare that with the sequential merge.

***Solution*:**

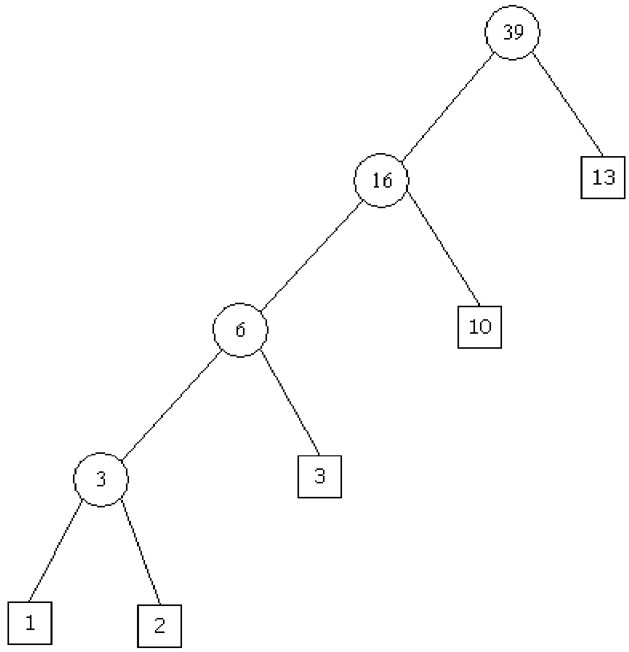


Cost = 4×3 + 4×1 + 3×10 + 2×3 + 1×13

= 8 + 4 + 30 + 6 + 13

= 61

Optimal merge: Sort the jobs { 1, 2, 3, 10, 13 }



Cost = 4×1 + 4×2 +3×3 + 2×10 + 1×13

= 4 + 8 + 9 + 20 + 13

54

∴ Optimal Cost is 54.

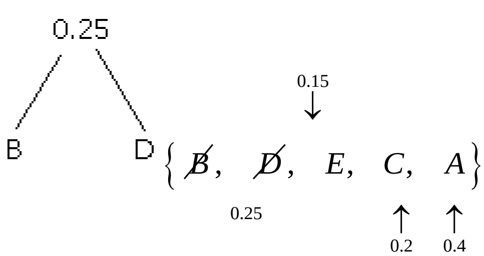
11.12 Construct Huffman codes for the following data:

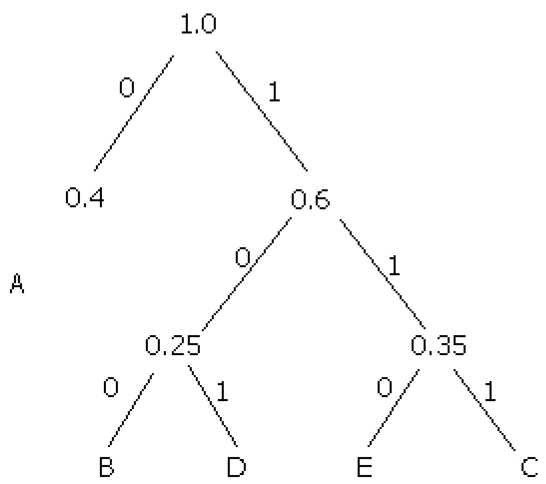
A	B	C	D	E
0.4	0.1	0.2	0.15	0.15

***Solution*:**

Sort the items in ascending order

{ ~~B~~, ~~D~~, E, C, A }





∴ A = 1

B = 1 0 0

D = 1 0 1

E = 1 1 0

C = 1 1 1

The encoded text

A B A C A E A D

Would be

{ 1 100 1 111 1 100 1 101 }

Decode text would	100	110	111	101	110
	B	*E*	*C*	*D*	*E*

The efficiency of Huffman code is = Entropy/ Average length of Huffman code

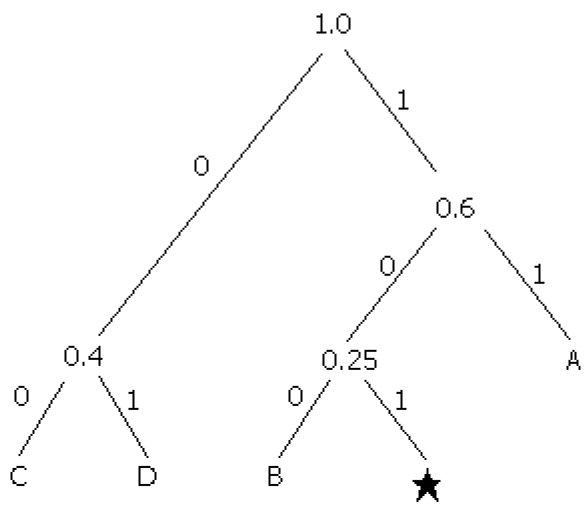
= 1.9219/2.6=0.7392 = 73.92%

11.13 Consider the following characters { A, B, C, D, * } with the following probability.

A	B	C	D	★
0.35	0.1	0.2	0.2	0.15

Show the construction of a Huffman tree and also show its efficiency.

Solution:

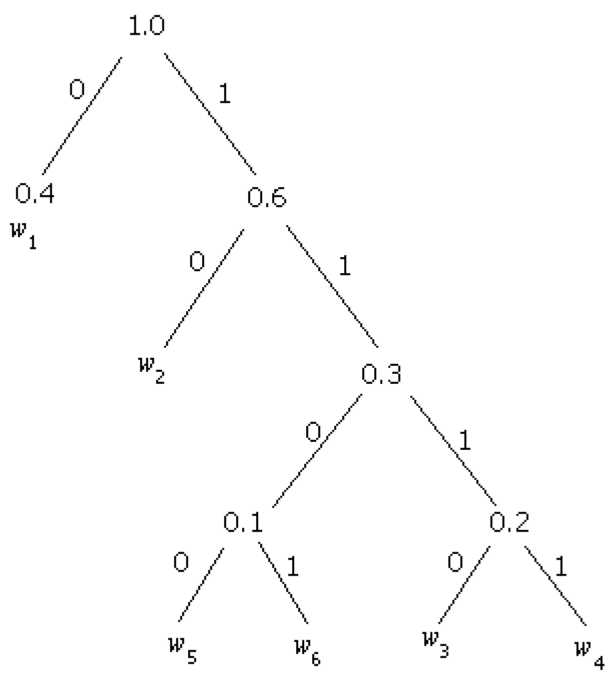


∴ C = 00 A = 11 B = 100

D = 01 ★ = 101

11.14 Construct a Huffman code for the following set of symbols.

w_1	w_2	w_3	w_4	w_5	w_6
0.4	0.3	0.1	0.1	0.05	0.05



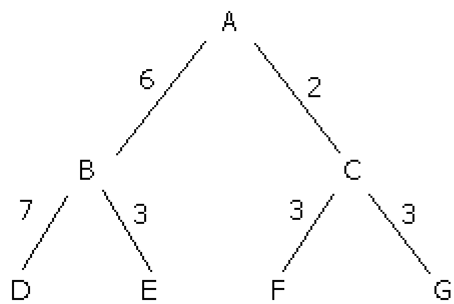
$w_1 = 0$ $w_3 = 1110$ $w_5 = 1100$

$w_2 = 10$ $w_4 = 1111$ $w_6 = 1101$

The efficiency of Huffman code is = Entropy/ Average length of Huffman code

= 1.9183/2.2 = 0.8726 = 87.26%

11.15 Consider the following weighted tree. Show the result of the tree vertex problem.

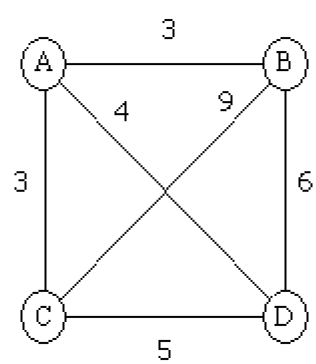


If the delay is 6, show the result of the tree vertex split algorithm.

***Solution*:**

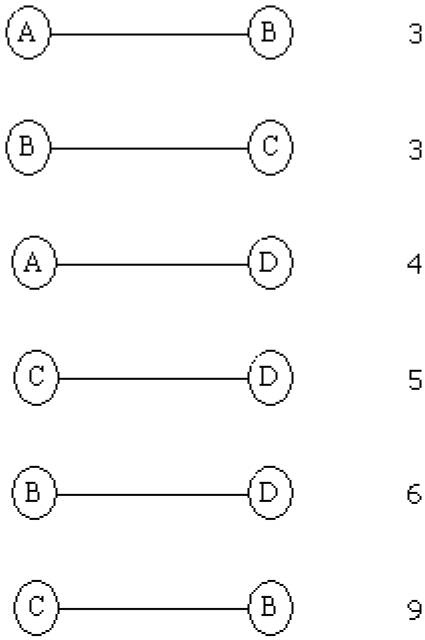
Hint: The split is in node B.

11.16 Use Kruskal's algorithm and construct an MST for the following graph:

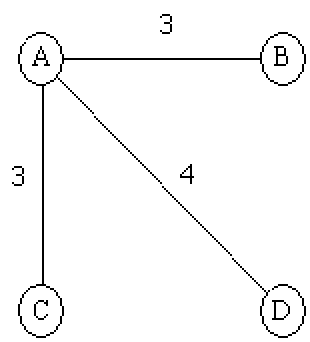


***Solution*:**

Sort the edges based on weight.



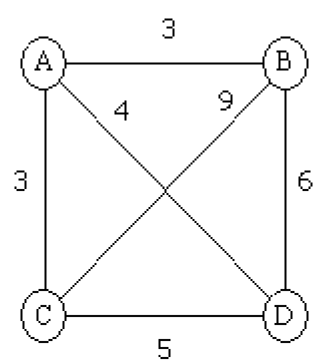
Add the edges such that there is no loop.



(C, D), (B, D), (C, B) are not possible.

11.17 Use Kruskal and Prim algorithm and construct an MST for the following graph:

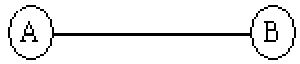
a)



Kruskal's algorithm is demonstrated in the previous problem

<u>Here using Prim's Algorithm:</u>

Let the starting edge be A—B

∴ 

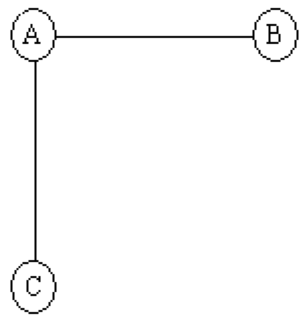
From A the possibilities as A—C (3)

B—D (4)

And from B possibilities as B—D (6)

B—C (9)

∴ The minimum edge is AC



Now the possibilities are

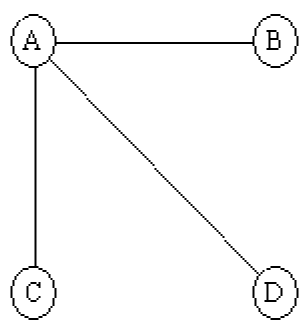
A D

C D

B D

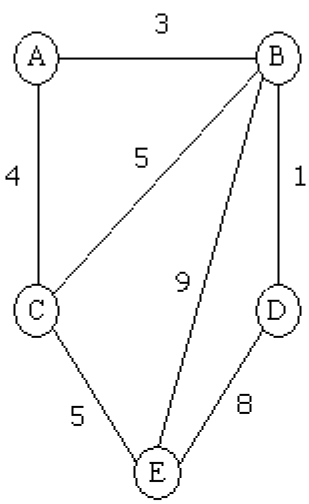
The minimum edge is A D.

∴ The MST is



All other edges are not possible and hence excluded.

b)

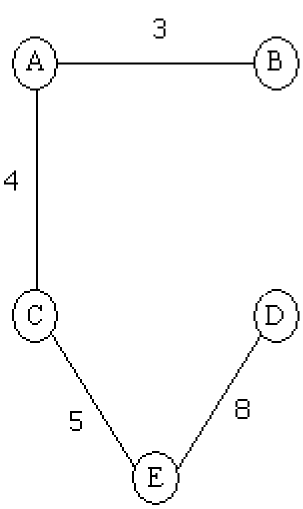


Using Kruskal's algorithm:

Let us sort the edges as per the cost as

(A, B)	3
(A, C)	4
(C, E)	5
(B, C)	5
(D, E)	8
(B, E)	9
(B, D)	11

Let us construct MST by selecting the least cost edges.



(B, C) Can't be chosen

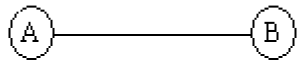
(B, E) Can't be chosen

(B, D) Can't be added as it creates loop

∴ The cost is 26.

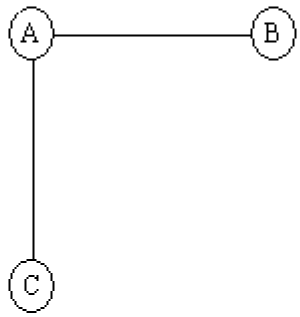
Prims Algorithm:

Let the starting point be



From node A, the possibilities are (A, E), (B, E) and (B, D) the minimum edge is 4.

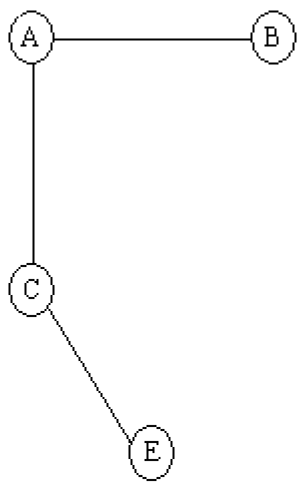
∴ New the MST would appear like



Now the possibilities are (B, E), (B, D), (C, B), (C, E) .

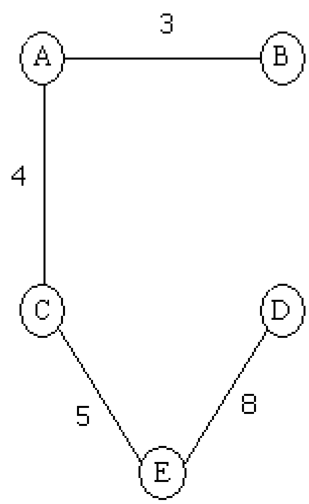
The minimum edge is (C, E).

Now the MST would appear like



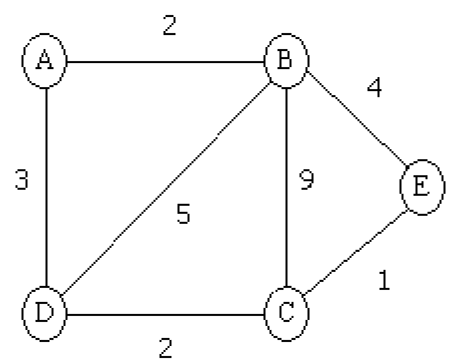
Now the possibilities are (B, C), (B, D), (C, B), (E, D).

Now the MST would appear like



∴ The cost of MST is 20.

11.18Apply Dijkstra's shortest path algorithm for the following graph and find the shortest path between vertices A and E.



In the next stop, node C would be added.

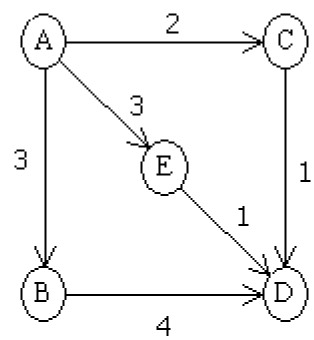
Initial {A}	V = A	V = B	V = C	V = D	V = E	Choosen Vertex
{A}	–	(2,B)	∞	(3,D)	∞	B
{A,B}	–	–	(9,C)	(3,A)	∞	D
{A,B,D}	–	–	(5,A)	–	(4,B)	E
	As the destination is E, one can stop here.					
{A,B,D,E}	–	–	(5,A)	–	–	C

The shortest path would be

A ⟶ B ⟶ E.

There is another shortest path, i.e., A—D—C—E whose length is also 6.

11.19Apply Dijkstra's algorithm for the following digraph and find the path between nodes A and D.



Initial {}	V = A	V = B	V = C	V = D	V = E	Choosen Vertex
{A}	–	(5,A)	(2,A)	∞	∞	{C}
{A,C}	–	(5,A)	–	(1,C)	(3,A)	{D}
{A,C,D}	–	(5,A)	–	–	(A,E)	{E}
{A,C,D,E}	–	(5,A)	–	–	–	{B}
{A,C,D,E,D}						

Since the destination is E, are one can stop with that vertex itself giving the shortest path

A ⟶ C ⟶ D with the shortest length 3.