TA: Jade Cheng

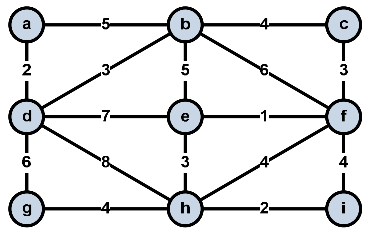
ICS 241

Recitation Lecture Notes #12

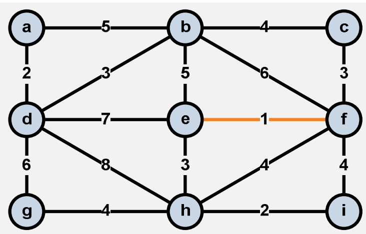
November 13, 2009

Recitation #12

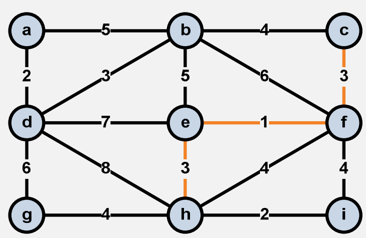
Question: Use Prim's algorithm to find a minimum spanning tree for the given weighted graph.



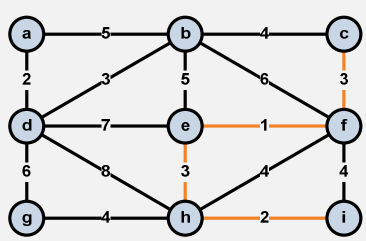
Answer: Step 1. Start from the smallest weight edge $\{e, f\}$



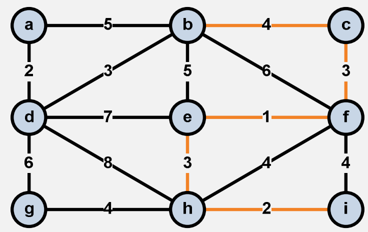
Step 2, 3. Append the smallest edge incidence with $\{e, f\}$, if the resulting graph does not contain any simple circuit. Then append to $\{e, f\}\{c, f\}$, or$\{e, f\}, \{e, h\}$ the smallest edge. You'll find the order of $\{c, f\}$ and $\{e, h\}$ doesn't matter.



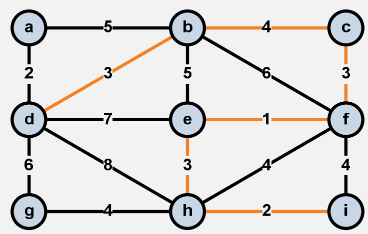
Step 4. Append the smallest edge incidence with $\{e, f\}, \{c, f\}, \{e, h\}$, if the resulting graph does not contain any simple circuit.



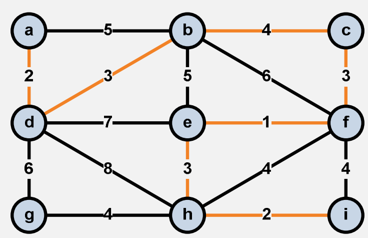
Step 5. Append the smallest edge incidence with $\{e, f\}, \{c, f\}, \{e, h\}, \{h, i\}$, if the resulting graph does not contain any simple circuit.



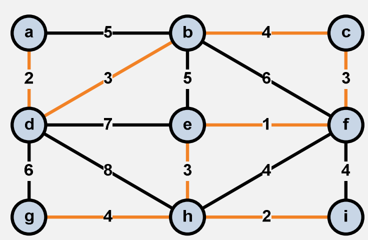
Step 6. Append the smallest edge incidence with $\{e, f\}, \{c, f\}, \{e, h\}, \{h, i\}, \{b, c\}$, if the resulting graph does not contain any simple circuit.



Step 7. Append the smallest edge incidence with $\{e, f\}, \{c, f\}, \{e, h\}, \{h, i\}, \{b, c\}, \{b, d\}$, if the resulting graph does not contain any simple circuit.

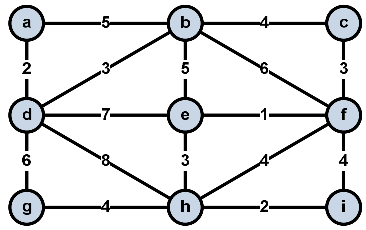


Step 8. Append the smallest edge incidence with $\{e, f\}, \{c, f\}, \{e, h\}, \{h, i\}, \{b, c\}, \{b, d\}, \{a, d\}$, if the resulting graph does not contain any simple circuit.

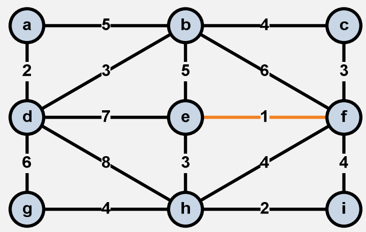


At this point all vertices are traversed. Prim's algorithm resulted a minimum spanning tree with a total weight of 22, and vertices $\{e, f\}, \{c, f\}, \{e, h\}, \{h, i\}, \{b, c\}, \{b, d\}, \{a, d\}, \{g, h\}$.

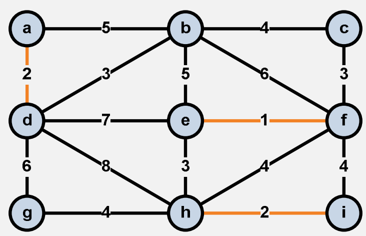
Question: Use Kruskal's algorithm to find a minimum spanning tree for the given weighted graph.



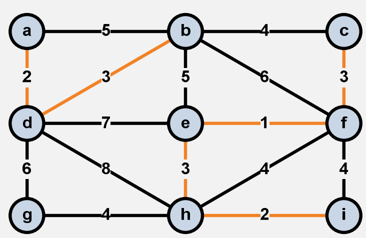
Answer: Step 1. Start from the smallest weight edge $\{e, f\}$.



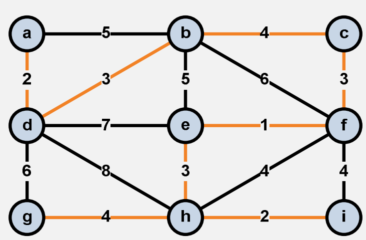
Step 2. Append the smallest edges in the rest of the graph, if the resulting graph doesn't contain any simple circuit. Now we have $\{e, f\}, \{a, d\}, \{h, i\}$.



Step 3. Append the smallest edges in the rest of the graph, if the resulting graph doesn't contain any simple circuit. Now we have $\{e, f\}, \{a, d\}, \{h, i\}, \{b, d\}, \{e, h\}, \{c, f\}$.



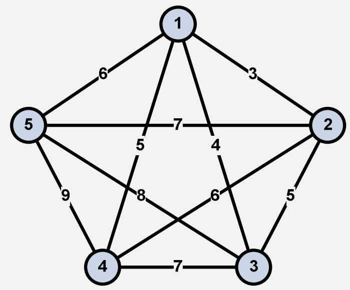
Step 4. Append the smallest edges in the rest of the graph, if the resulting graph doesn't contain any simple circuit. Now we have $\{e, f\}, \{a, d\}, \{h, i\}, \{b, d\}, \{e, h\}, \{c, f\}, \{g, h\}, \{b, c\}$.



At this point all vertices are traversed. Kruskal's algorithm resulted a minimum spanning tree with a total weight of 22. For this example, the minimum spanning tree resulted by these two different algorithms happen to be the same. This is not always the case, in fact, most of the time, they are different. But the total weight of the minimum spanning tree would always be the same.

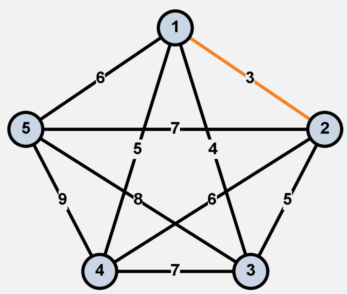
Question: Suppose the vertices of K_5 are numbered $1, 2, 3, 4, 5$ and each edge is assigned a weight equal to the sum of the labels on the endpoints of the edge. Find a spanning tree of minimum weight for this graph.

Answer: First, we need to construct this weighted simple connected graph and use it as our input for finding the minimum spanning tree.

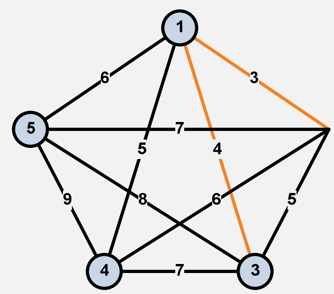


Then we need to construct a minimum spanning tree and count the total weight, which will be the answer of this problem. If we use Kruskal's algorithm:

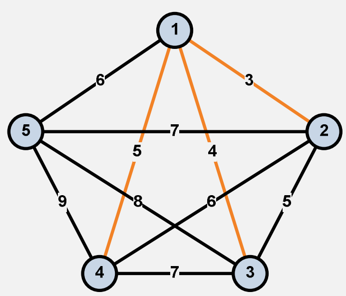
Step 1. $\{1, 2\}$



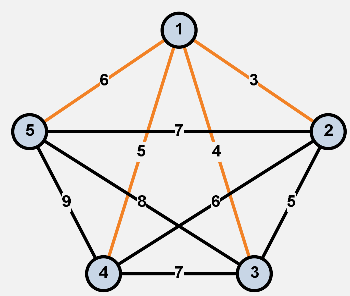
Step 2. $\{1, 2\}, \{1, 3\}$



Step 3. $\{1, 2\}, \{1, 3\}, \{1, 4\}$



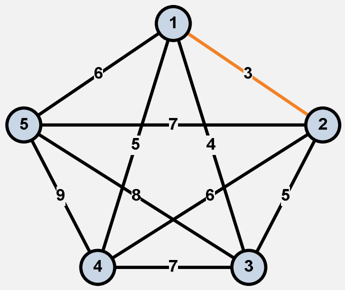
Step 4. $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}$

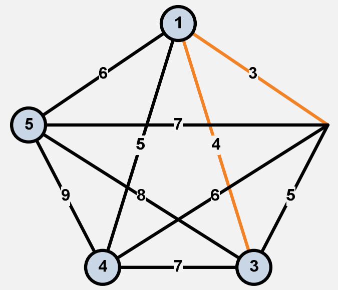


At this point all vertices are traversed and the algorithm terminates. The resulting minimum spanning tree has a total weight of $3 + 4 + 5 + 6 = 18$. We can also use Prim's algorithm:

Step 1. $\{1, 2\}$

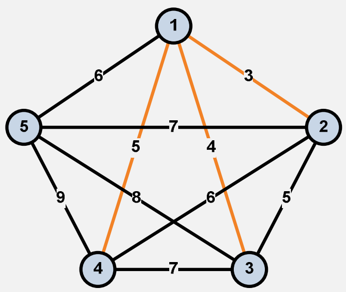
Step 2. Incidence to $\{1, 2\}$, append $\{1, 3\}$

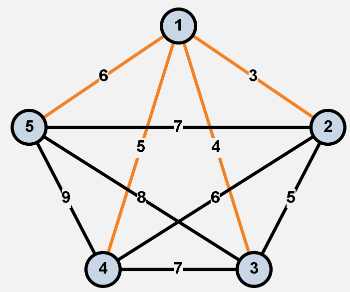




Step 3. Incidence to $\{1,2\},\{1,3\}$, append $\{1,4\}$

Step 4. Incidence to $\{1,2\},\{1,3\},\{1,4\}$, append $\{1,5\}$





At this point all vertices are traversed and the algorithm terminates. It happen to be the same tree as the Kruskal's algorithm above. The minimum total weight of the graph is 18.

Question: Suppose the vertices of K_n are numbered $1, 2, \cdots, n$ (in clockwise order) and each edge is assigned a weight equal to the sum of the labels on the endpoints of the edge. Find a spanning tree of minimum weight for this graph and find the weight of this spanning tree.

Answer: The spanning tree of minimum cost has edges $\{1,2\},\{1,3\},\ldots,\{1,n\}$.

Using either Kruskal's Algorithm or Prim's Algorithm, the first edges added are $\{1,2\}$ and $\{1,3\}$. At the next stage, edges $\{2,3\}$ and $\{1,4\}$ have the smallest weight, but adding edge $\{2,3\}$ would create a circuit (circuit containing$\{2,3\},\{1,2\},\{1,3\}$). Therefore edges $\{1,2\}, \{1,3\},$ and $\{1,4\}$ are inserted into the spanning tree. At the next stage, edges $\{2,4\}$ and $\{1,5\}$ have the smallest weight, but adding edge $\{2,4\}$ would create a circuit (circuit containing$\{2,4\},\{1,2\},\{1,4\}$). Therefore edges $\{1,2\}, \{1,3\}, \{1,4\},$ and $\{1,5\}$ are inserted into the spanning tree.

In general, if edges $\{1,2\},\{1,3\},\cdots,\{1,k\}$ have been selected, the next edge inserted must be $\{1,k+1\}$ (of weight $k+2$). (Any other edge $\{i,j\}$ with weight $\leq k+2$ would have $1 < i \leq k$ and $1 < j \leq k$ and would create a circuit when combined with $\{1,i\}$ and $\{1,j\}$.) Thus, the spanning tree of minimum weight consists of $\{1,2\},\{1,3\},.\cdots,\{1,n\}$.

Its total weight is, therefore,

$$(1+2)+(1+3)+(1+4)+\cdots+(1+n)$$

$$=(n-1)+(2+3+4+\cdots+n)$$

$$=(n-1)+\frac{(n+2)\cdot(n-1)}{2}$$

$$=\frac{(n+4)\cdot(n-1)}{2}$$

Question: Find the values of these expressions

Review: The Boolean sum, denoted by $+$ or *OR*, has the following values:

$1+1=1$ $\quad$ $1+0=1$ $\quad$ $0+1=1$ $\quad$ $0+0=0$

The Boolean product, denoted by $\cdot$ or by *AND*, has the following values:

$1\cdot 1=1$ $\quad$ $1\cdot 0=0$ $\quad$ $0\cdot 1=0$ $\quad$ $0\cdot 0=0$

a. $1\cdot\bar{0}$

Answer: $1\cdot\bar{0}=1\cdot 1=1$.

b. $1+\bar{1}$

Answer: $1+\bar{1}=1+0=1$.

c. $\bar{0}\cdot 0$

Answer: $\bar{0}\cdot 0=1\cdot 0=0$.

d. $\overline{(1+0)}$

Answer: $\overline{(1+0)}=\bar{1}=0$.

Question: Show that $(1\cdot 1)+(\overline{0\cdot 1}+0)=1$

Answer: Following the Boolean operations reviewed above, we can derive the left into the right side, 1.

$$(1\cdot 1)+(\overline{0\cdot 1}+0)=1+(\bar{0}+0)$$

$$=1+(1+0)$$

$$=1+1$$

$$=1.$$

Question:	Translate the equation in the previous question into a propositional equivalency by changing each 0 to F, each 1 to a T, each Boolean sum into a disjunction, each Boolean product into a conjunction, each complementation into a negation, and the equals sing to a propositional equivalence sign.
Answer:	$(1 \cdot 1) + (\overline{0 \cdot 1} + 0) = (T \wedge T) \vee (\neg(F \wedge T) \vee F) \equiv T$

Question:	Prove the idempotent law $x = x \cdot x$ using the other identities of Boolean algebra listed in Table 5 in the textbook

Review:	$\bar{\bar{x}} = x$	Law of the double complement
	$x + x = x$ and $x \cdot x = x$	Idempotent laws
	$x + 0 = x$ and $x \cdot 1 = x$	Identity laws
	$x + 1 = 1$ and $x \cdot 0 = 0$	Domination laws
	$x + y = y + x$ and $xy = yx$	Commutative laws
	$x + (y + z) = (x + y) + z$ and $x(yz) = (xy)z$	Associative laws
	$x + yz = (x + y)(x + z)$ and $x(y + z) = xy + xz$	Distributive laws
	$\overline{(xy)} = \bar{x} + \bar{y}$ and $\overline{(x + y)} = \bar{x}\bar{y}$	De Morgan's laws
	$x + xy = x$ and $x(x + y) = x$	Absorption laws
	$x + \bar{x} = 1$	Unit property
	$x\bar{x} = 0$	Zero property

Answer: Follow the Boolean algebra operations we have,

$x = x \cdot 1$	identity law
$= x \cdot (x + \bar{x})$	unit property
$= x \cdot x + x \cdot \bar{x}$	distributive law
$= x \cdot x + 0$	zero property
$= x \cdot x$	identity law

Question:	Prove the idempotent law $x \cdot 0 = 0$ using the other identities of Boolean algebra listed in Table 5 in the textbook

Answer: Follow the Boolean algebra operations we have,

$x \cdot 0 = x \cdot (x \cdot \bar{x})$	zero property
$= (x \cdot x) \cdot \bar{x}$	associative law
$= x \cdot \bar{x}$	idempotent law
$= 0$	zero property

Question: Use a table to express the values of each of these Boolean function.

a. $F(x, y, z) = \bar{x}y$

Answer:

x	y	z	$\bar{x}$	$\bar{x}y$
1	1	1	0	0
1	1	0	0	0
1	0	1	0	0
1	0	0	0	0
0	1	1	1	1
0	1	0	1	1
0	0	1	1	0
0	0	0	1	0

b. $F(x, y, z) = \bar{x}y + \bar{y}z$

Answer:

x	y	z	$\bar{x}$	$\bar{y}$	$\bar{x}y$	$\bar{y}z$	$\bar{x}y + \bar{y}z$
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	1	0	1	0	1	1
1	0	0	0	1	0	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	0	1
0	0	1	1	1	0	1	1
0	0	0	1	1	0	0	0

Question: What values of the Boolean variables x, y satisfies $xy = x + y$?

Answer: We can solve this problem by looking over its membership table.

x	y	xy	$x + y$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

The first and last entry contain the same Boolean value for xy and $x + y$. Therefore $(1, 1)$ and $(0, 0)$ satisfy the condition.

Question: Determine whether or not the following identities hold.

a. $x \oplus y = (x \oplus y) \oplus z$

Answer: The symmetric difference $A \oplus B$ of two sets A and B is the set containing those elements in either A or B but not in both A and B.

x	y	z	$x \oplus y$	$(x \oplus y) \oplus z$
T	T	T	F	T
T	T	F	F	F
T	F	T	T	F
T	F	F	T	T
F	T	T	T	F
F	T	F	T	T
F	F	T	F	T
F	F	F	F	F

By looking at the last two columns, we see that the fourth and eighth entries have the different Boolean values for $x \oplus y$ and $(x \oplus y) \oplus z$. Therefore, the identity $x \oplus y = (x \oplus y) \oplus z$ does not always hold.

In this example, we have a short cut. The left side and right side share a component, $x \oplus y$. So, basically, the identity is saying the Boolean value z does not matter in the Boolean expression $(x \oplus y) \oplus z$. This is obviously not true. When $x \oplus y$ is False, $(x \oplus y) \oplus z$ would always be False, the value of z doesn't matter. But when $x \oplus y$ is True, and z is False, $(x \oplus y) \oplus z$ would be False instead of $x \oplus y = (x \oplus y) \oplus z =$ True.

b. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

Answer: The symmetric difference $A \oplus B$ of two sets A and B is the set containing those elements in either A or B but not in both A and B.

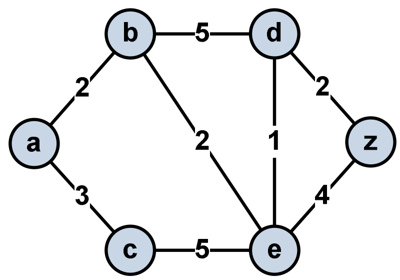
A	B	C	$A \oplus B$	$B \oplus C$	$A \oplus (B \oplus C)$	$(A \oplus B) \oplus C$
T	T	T	F	F	T	T
T	T	F	F	T	F	F
T	F	T	T	T	F	F
T	F	F	T	F	T	T
F	T	T	T	F	F	F
F	T	F	T	T	T	T
F	F	T	F	T	T	T
F	F	F	F	F	F	F

By looking at the last two columns, we see that the all of the entries have the same Boolean values for $A \oplus (B \oplus C)$ and $(A \oplus B) \oplus C$. Therefore, the identity $A \oplus (B \oplus C) = (A \oplus B) \oplus C$ always holds.

Note that we've proved that the symmetric difference operation $\oplus$ on sets is *associative*.

Dijkstra's Algorithm Review

Exercise 2: Find the length of a shortest path between a and z in the given weighted graph. [12 pts]



Answer: Let's start from the original graph:

Step 1:

$S = \emptyset$

$a = \infty$

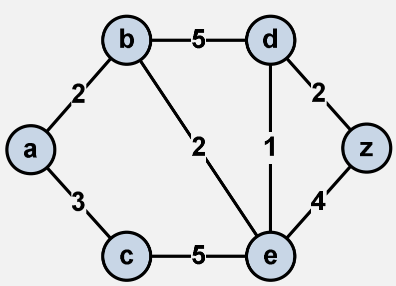
$b = \infty$

$c = \infty$

$d = \infty$

$e = \infty$

$z = \infty$



Step 2:

$S = \{a\}$

$a = 0$

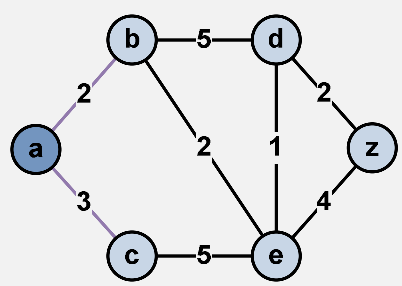
$b = 0 + 2 = 2$

$c = 0 + 3 = 3$

$e = \infty$

$e = \infty$

$z = \infty$



Step 3:

$S = \{a, b\}$

$a = 0$

$b = 2$

$c = 3$

$e = 2 + 2 = 4$

Step 4:

$S = \{a, b, c\}$

$a = 0$

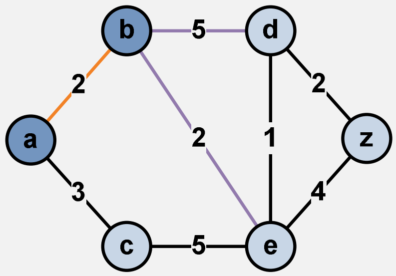
$b = 2$

$c = 3$

$e = 4$

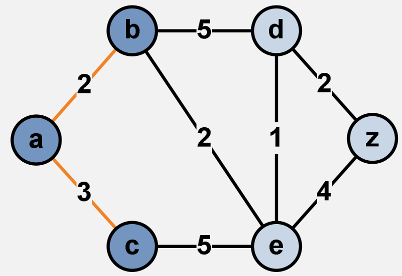
$d = 2 + 5 = 7$

$z = \infty$



$d = 7$

$z = \infty$



Step 5:

$S = \{a, b, c, e\}$

$a = 0$

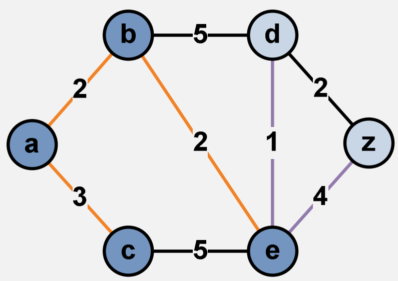
$b = 2$

$c = 3$

$e = 4$

$d = 4 + 1 = 5$

$z = 4 + 4 = 8$



Step 6:

$S = \{a, b, c, e, d\}$

$a = 0$

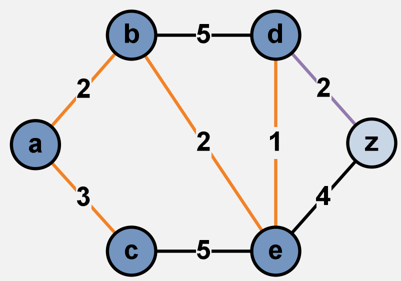
$b = 2$

$c = 3$

$e = 4$

$d = 5$

$z = 5 + 2 = 7$



Step 7:

$S = \{a, b, c, e, z\}$

$a = 0$

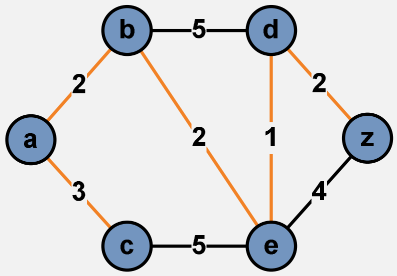
$b = 2$

$c = 3$

$e = 4$

$d = 5$

$z = 7$



At this point, vertex z, the destination is reached and appended to set S, therefore the main loop condition, $z \notin S$, can't be satisfied any more. Program exits the loop and terminates. The length is 7 and the path consists vertices: a, b, e, d, z in this order.