Greedy

Lecture 3

Assignment 3

Open : Sunday

Due : Satday

Nov 2

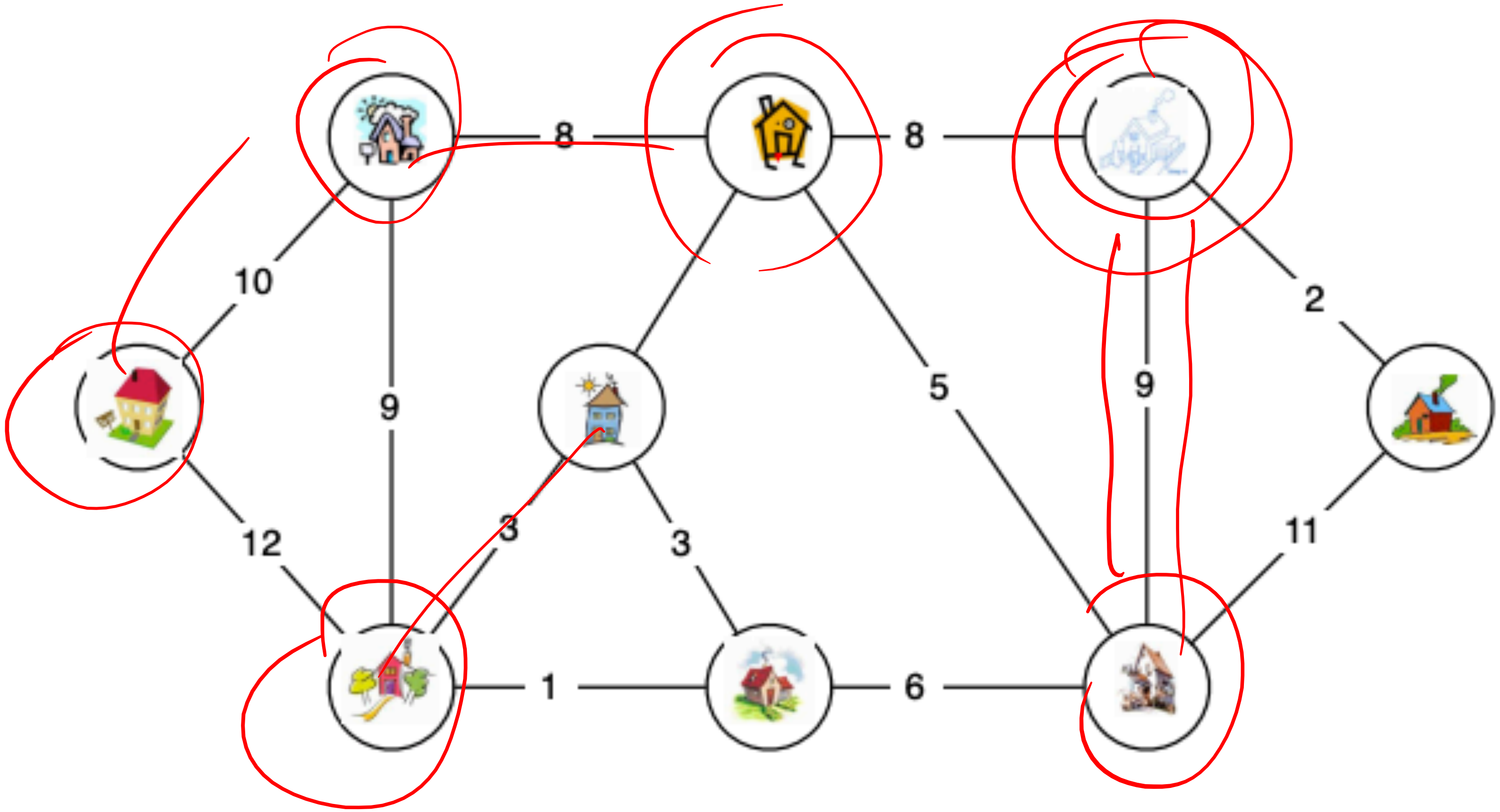
Connecting Houses

Connecting Houses

Connecting Houses

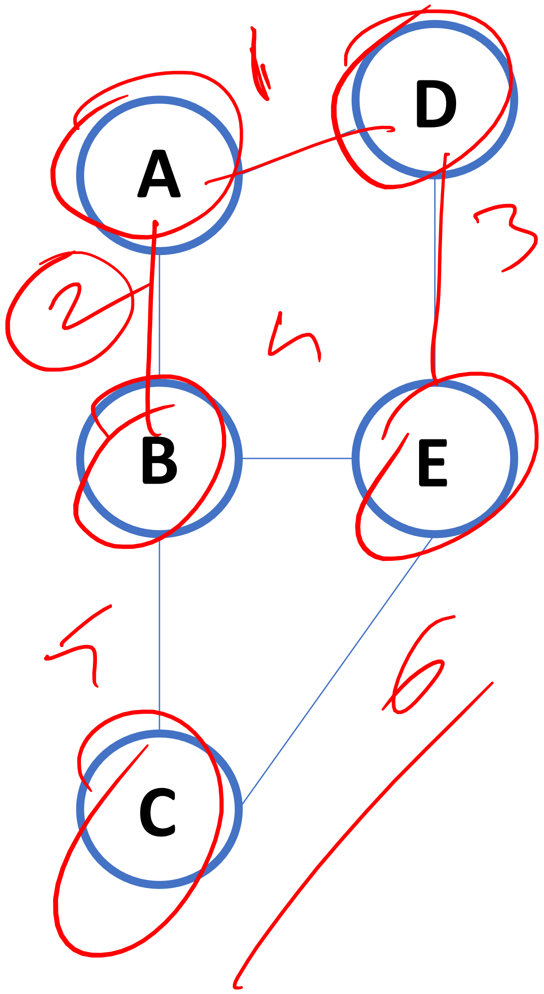
Connecting Houses

Connecting Houses



Graphs

G=(V,E)



V = vertices

E=edges

W(e)= weights for each edge, e ∈ E

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

E= {(A,B), (A,D), (B,C), (B,E), (C,E), (D,E)}

2 1

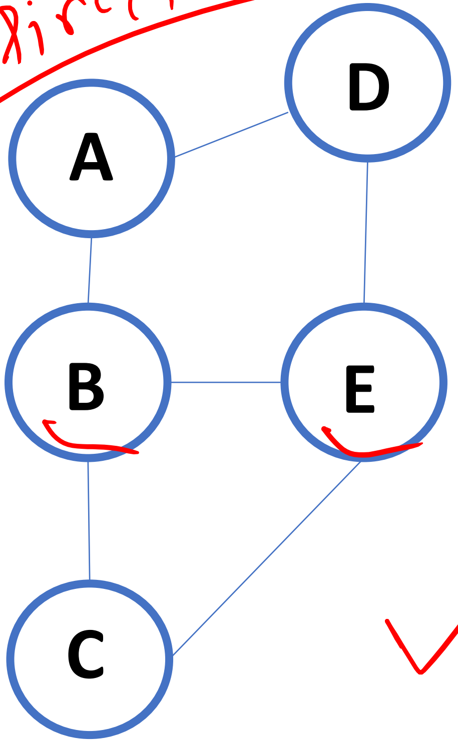
W(A,B) = 2

W(A,D) = 1

G=(V,E)

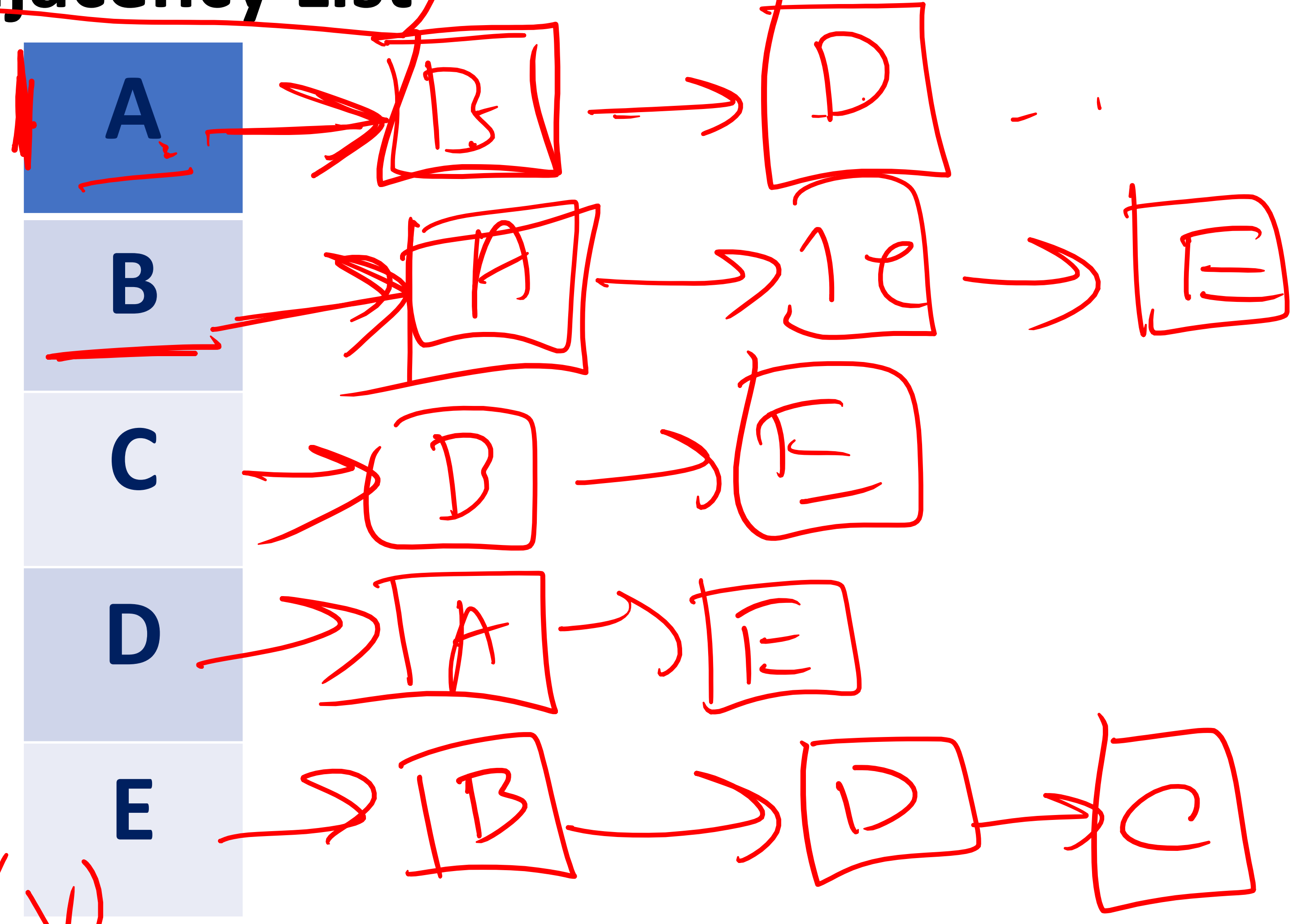
Non-direction

O(1)



Adjacency List

(A,B)

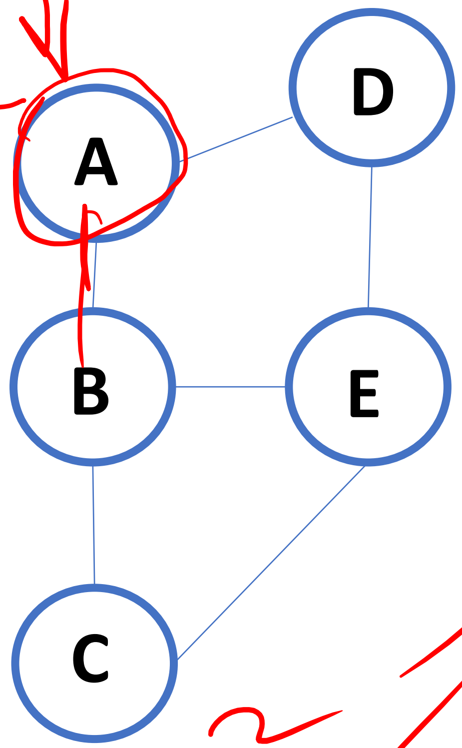


Space: |V|+|E|x2

Time list neighbors: factor(V)

Time to check the edges: factor(V)

G=(V,E)



Adjacency Matrix

(A, D)

	A	B	C	D	E
A	0	1	0	1	0
B	1	0	1	0	1
C	1	0	0	0	1
D	1	0	0	0	1
E	0	1	1	1	0

Space: V^2

Time list neighbors: fne(V)

Time to check the edges: $\boxed{O(1)}$

Path in a graph

v2 v3 (D,E) E E A D E B C

v1 v2 v3 v4 v5

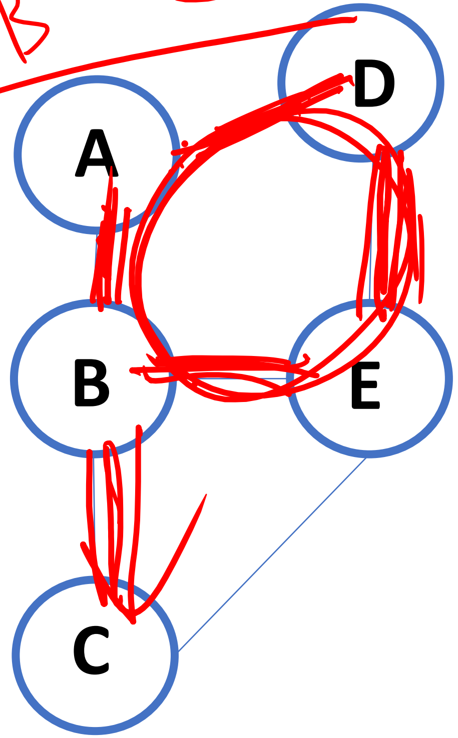
A D E B A D

E B C

- A sequence of nodes v1, v2, …., vk with the property that:

  $(v_i, v_{i+1}) \in E$. **for all i=1,….., k-1**

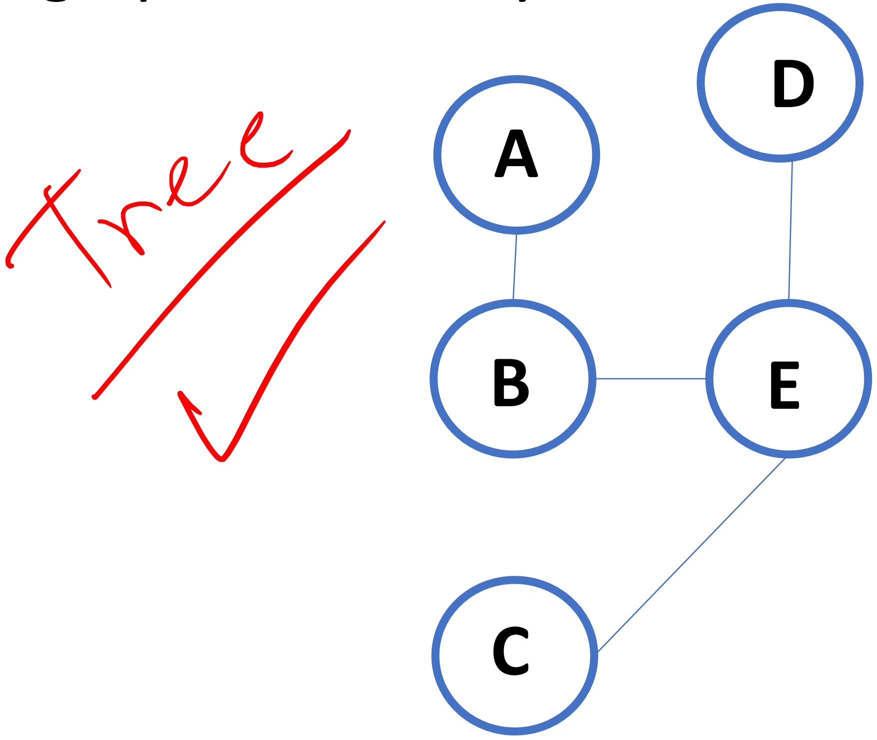
- **Simple path:** is a path where each vertex appears at most once
- **Cycle (path):** is a path of length > 2, such that there is atleast two vertex such that $v_i=v_k$

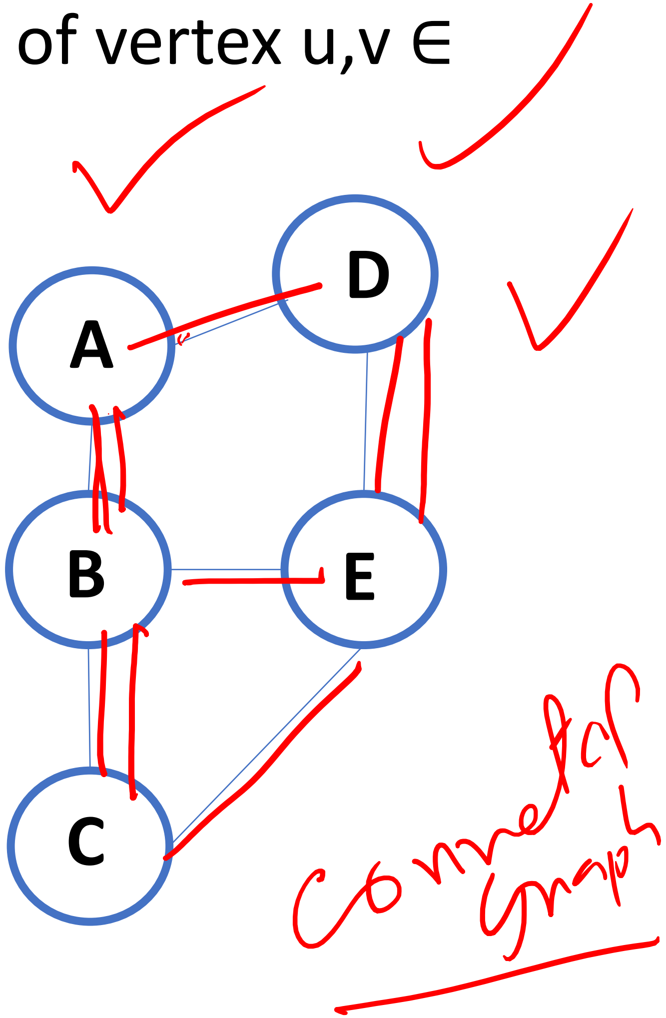


n = (E) A A D E B C

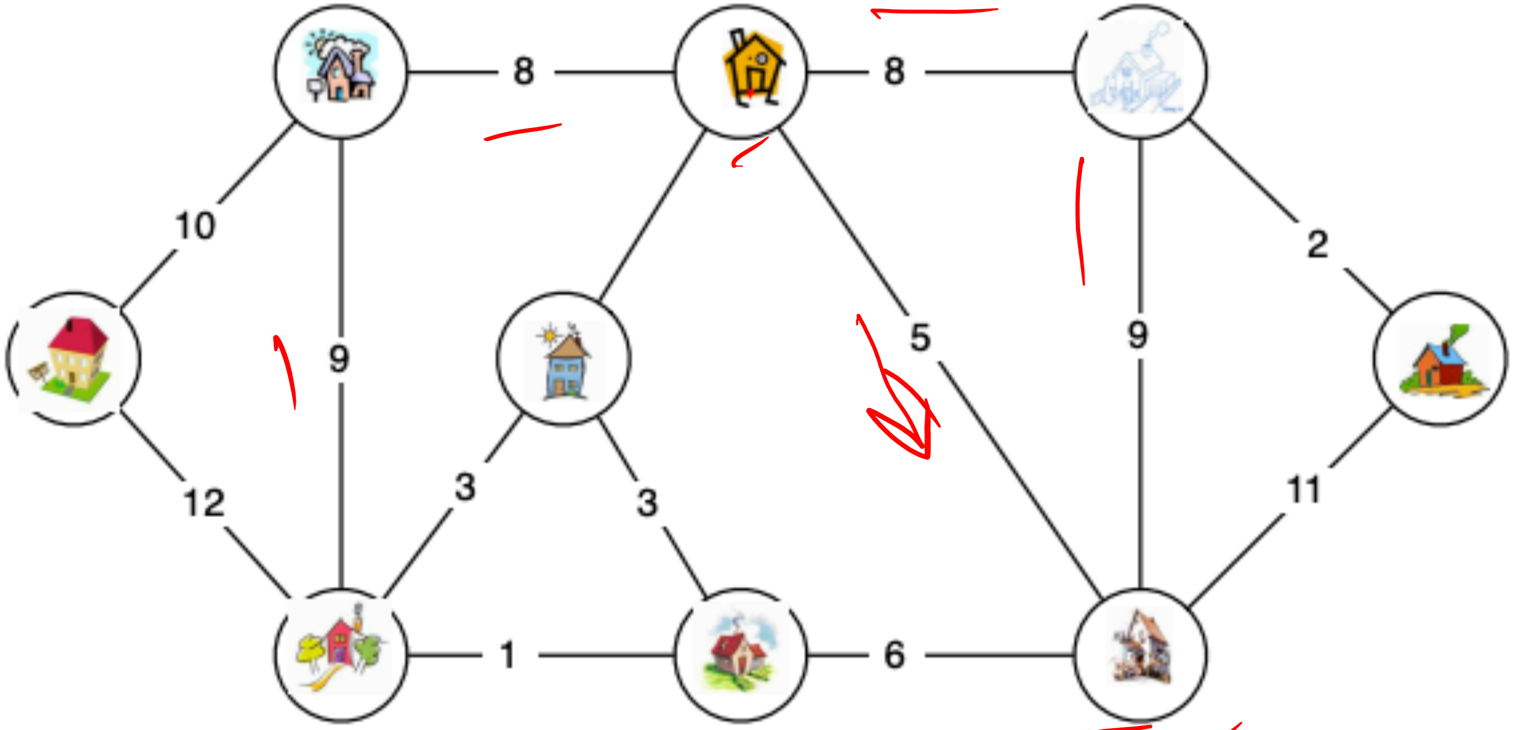
Tree

- **Connected graph**: is a graph G=(V,E) such that any pair of vertex u,v ∈ V, there exists a path from u to v.
- **A Tree:** is a connected graph with no cycle.





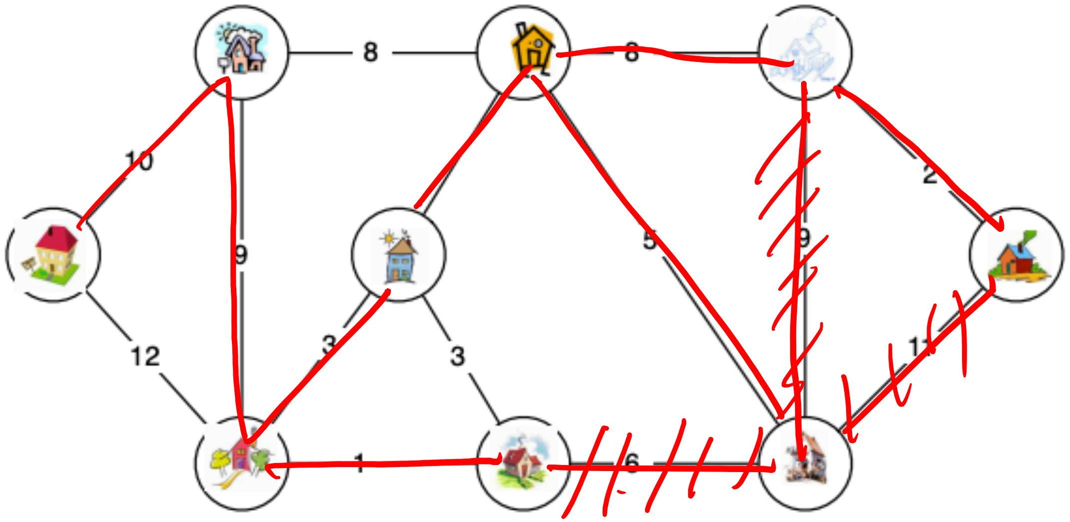
Minimum Spaning Tree



We want a tree that connects all **V**, of graph **G**, and has **minimum cost**

Minimum Spaning Tree

$\boxed{T} \subset \boxed{E}$

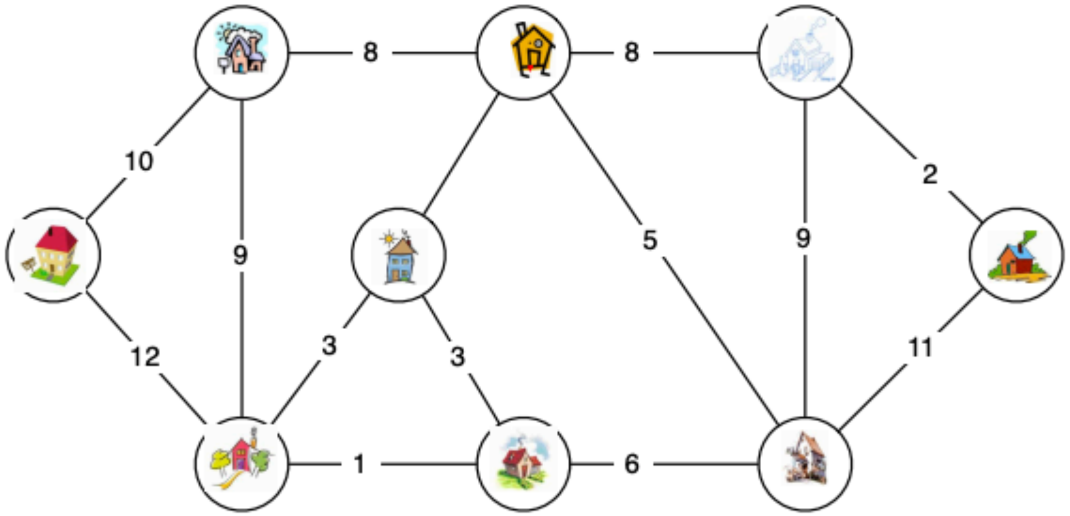


Looking for a set of edges such that T ⊆ E

1. Connects all vertices (V)
2. Has the least cost:

$$\text{Min} \sum_{(u,v) \in T} w(u,v)$$

Minimum Spaning Tree



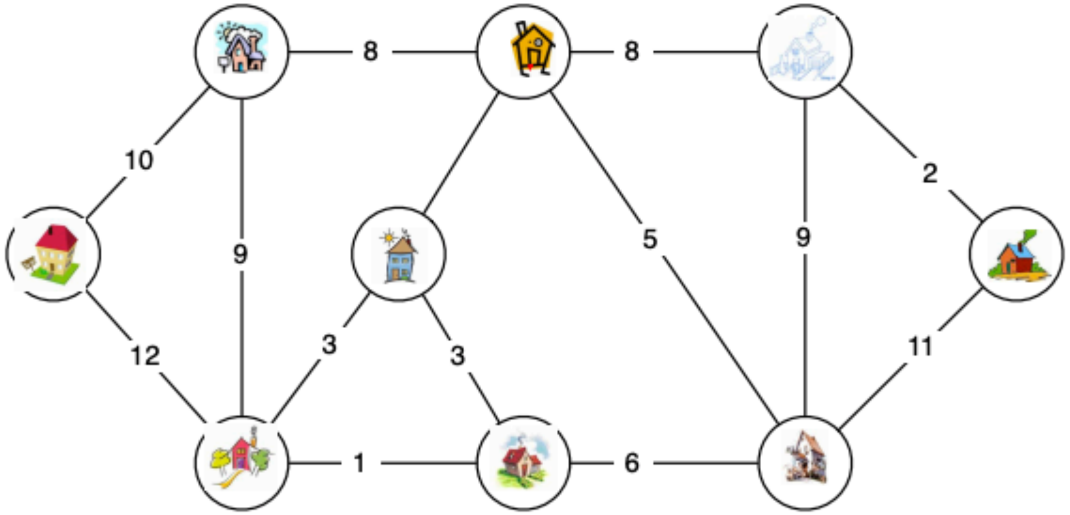
Looking for a set of edges such that T ⊆ E

1. Connects all vertices (V)
2. Has the least cost:

$$\text{Min} \sum_{(u,v)\in T} w(u,v)$$

How many edges does the solution have?

Minimum Spaning Tree



Looking for a set of edges such that T ⊆ E

1. Connects all vertices (V)
2. Has the least cost:

$$\text{Min} \sum_{(u,v)\in T} w(u,v)$$

How many edges does the solution have?

V-1

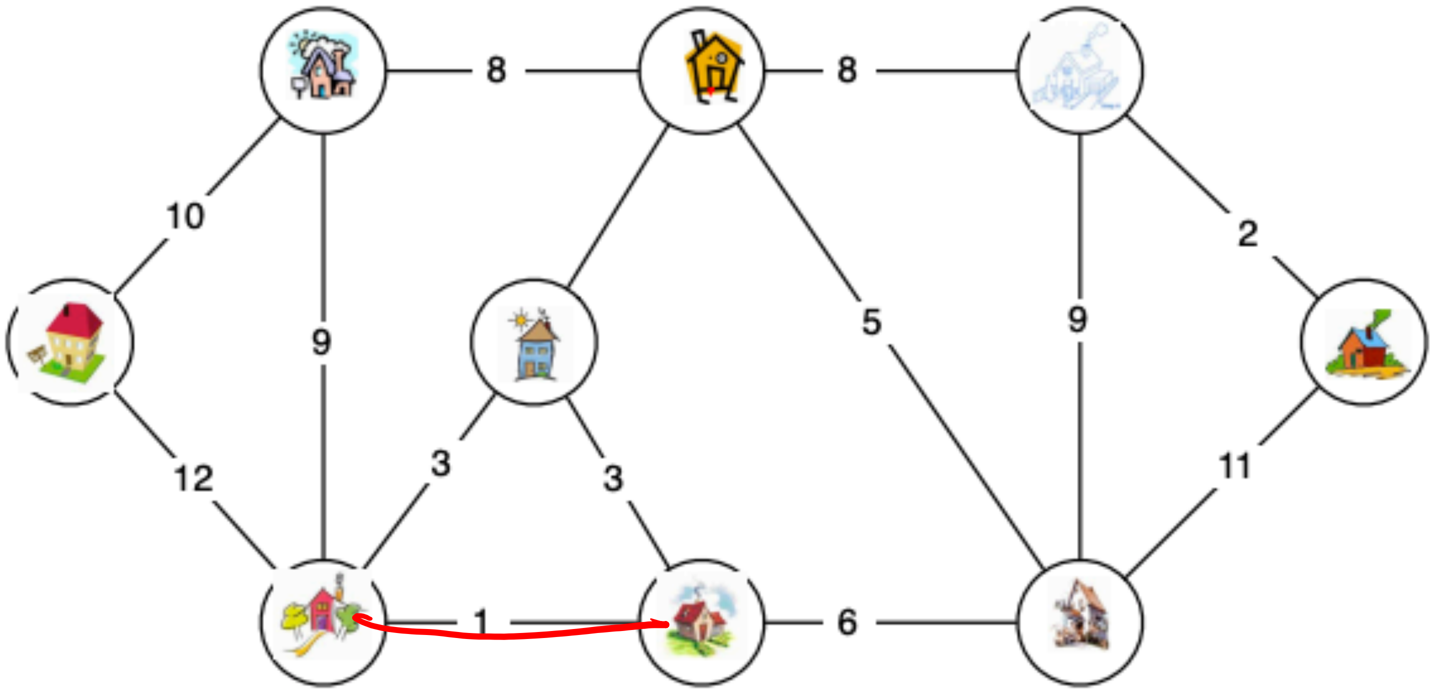
Does the solution have a cycle?

Strategy

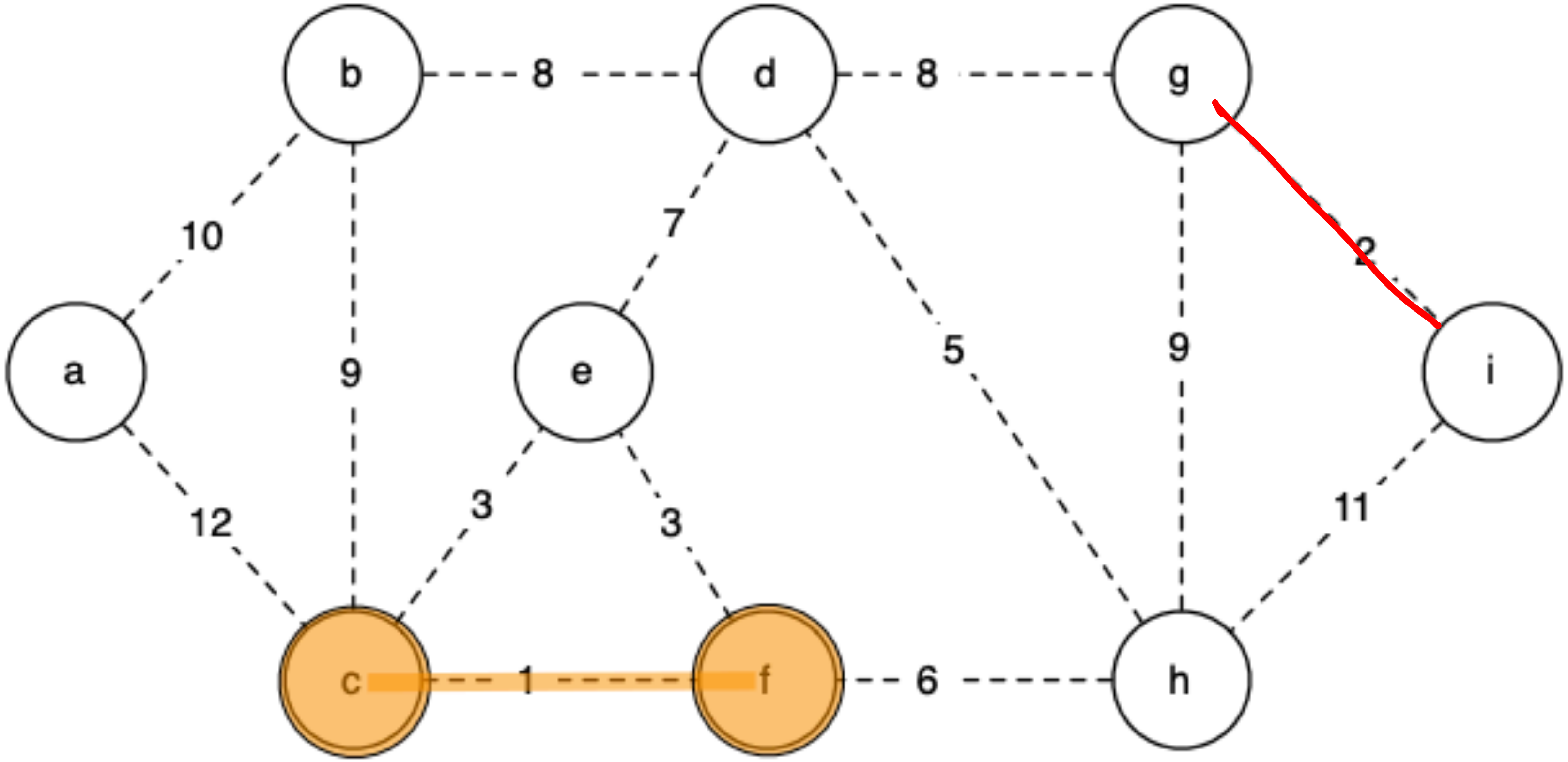
- Start with an empty set of edges A
- Repeat for v-1 times:
  - Add lightest edge that does not create a cycle

Example

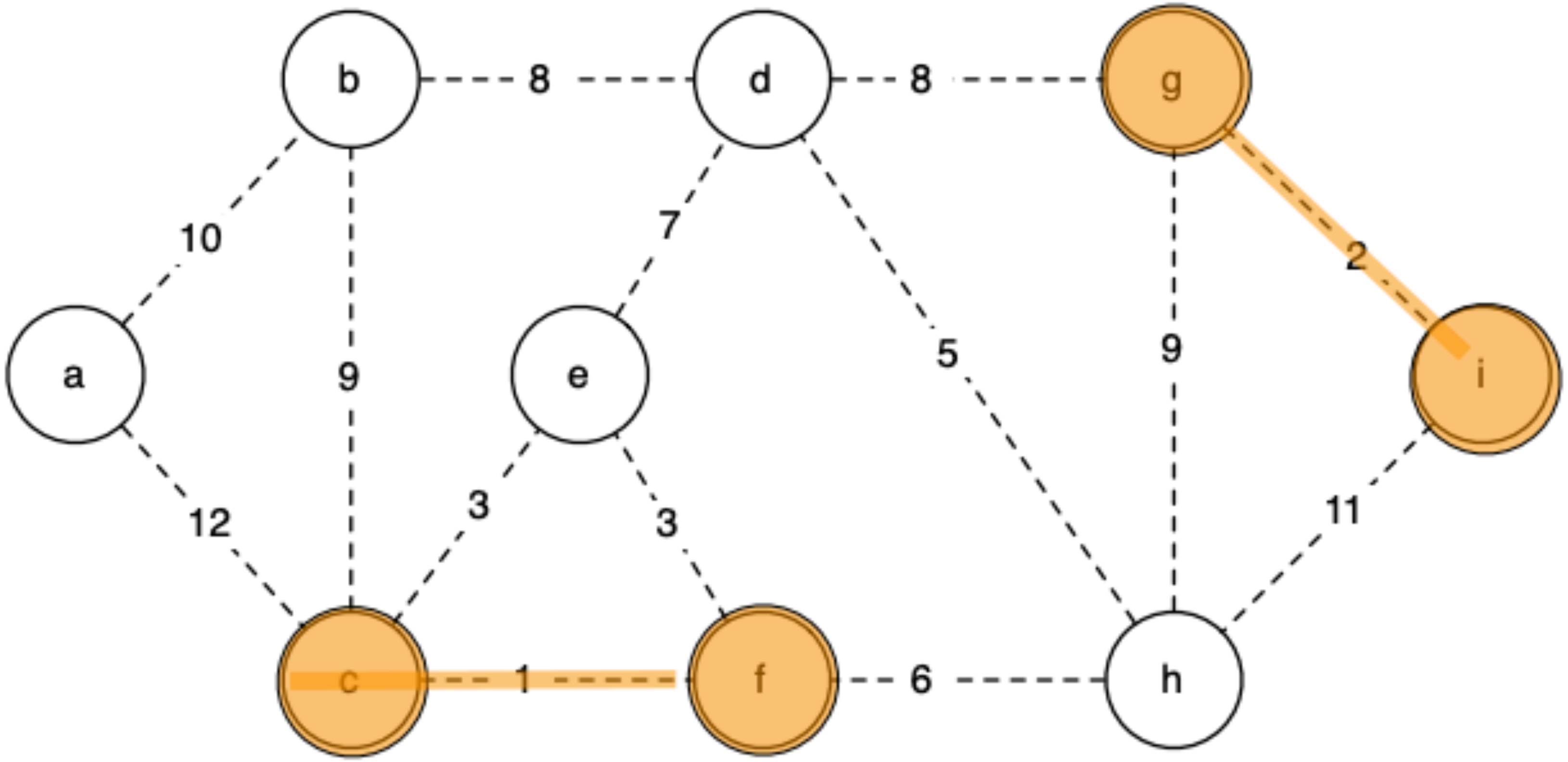
A= Ø



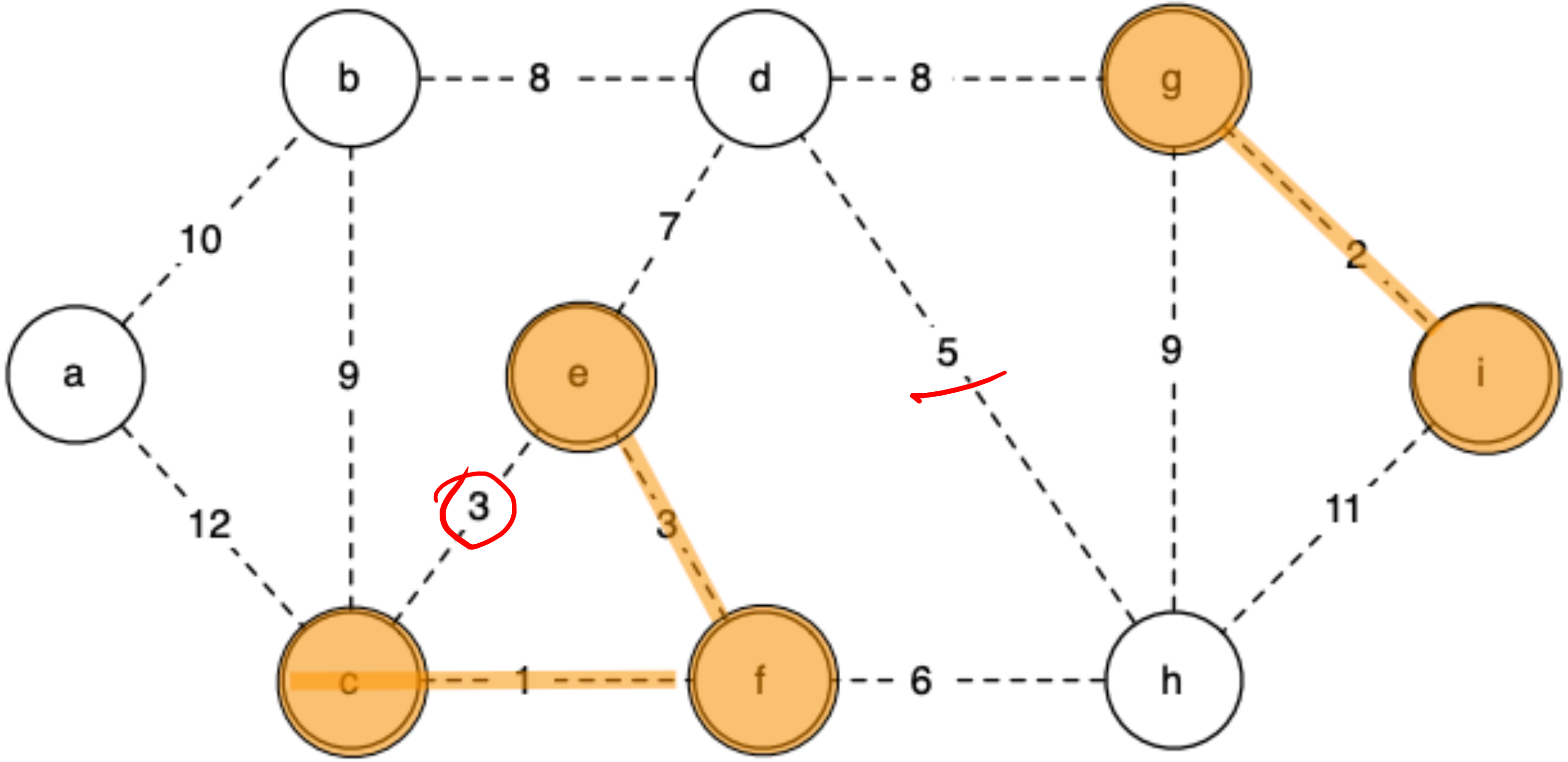
Example



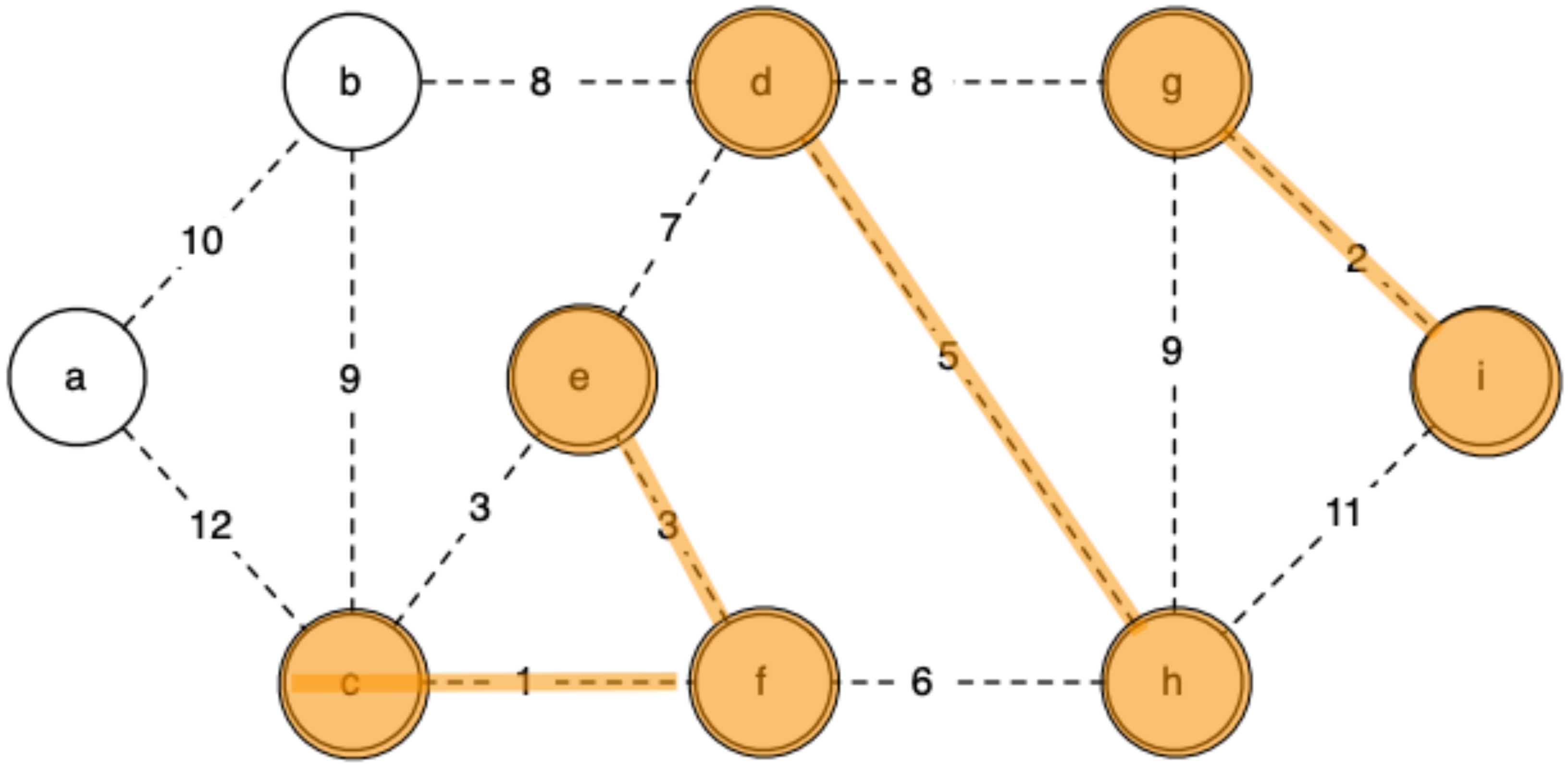
Example



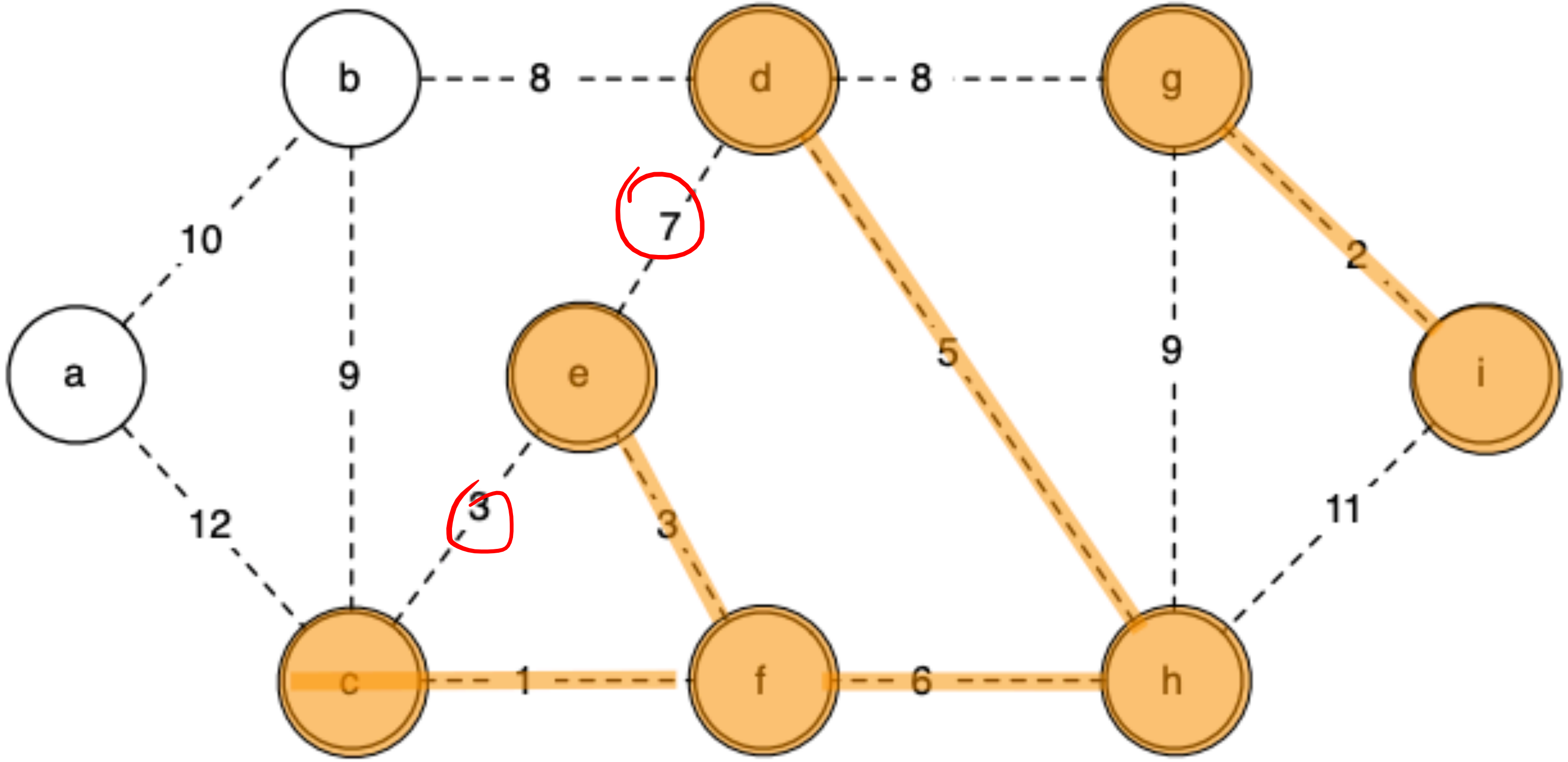
Example



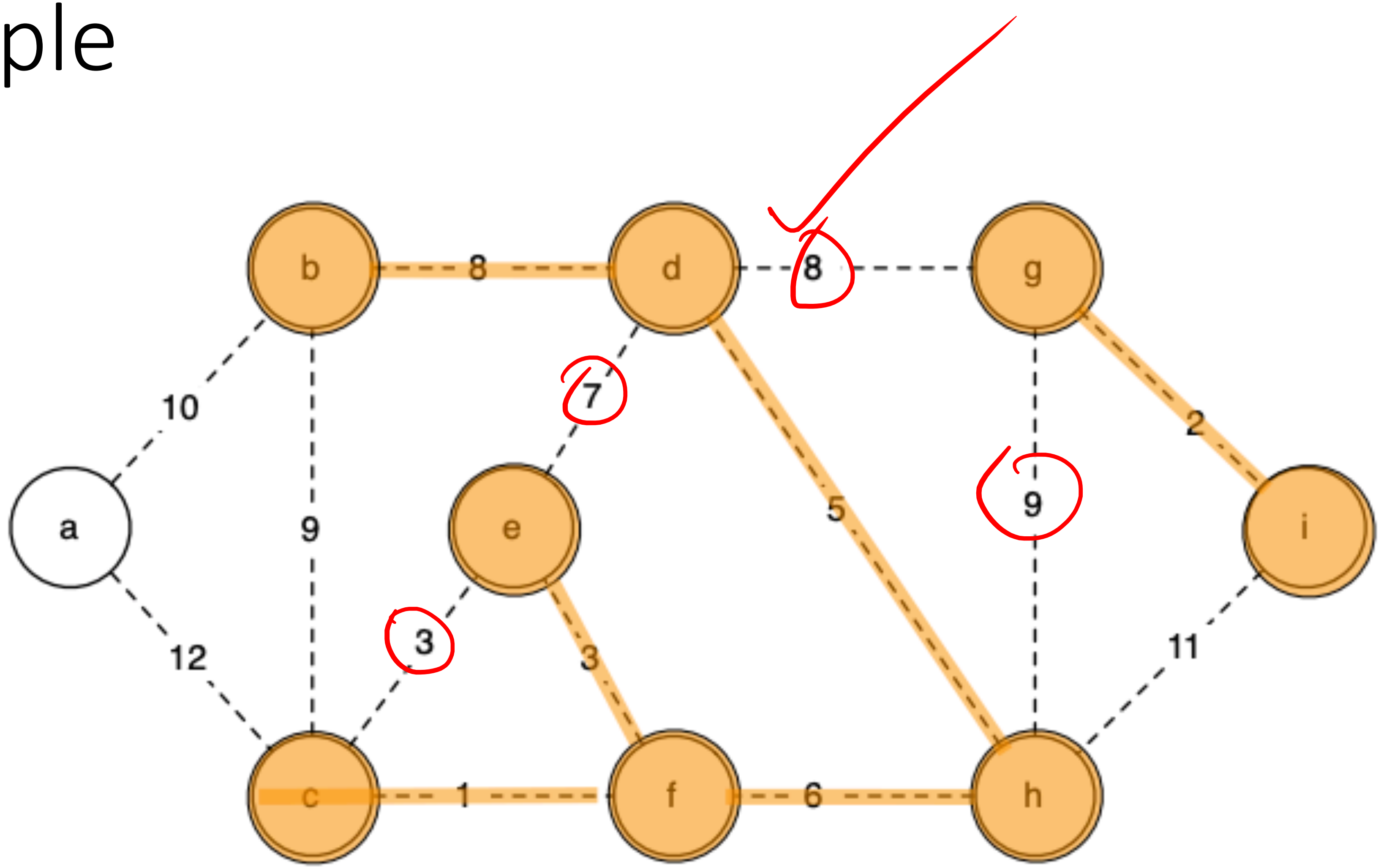
Example



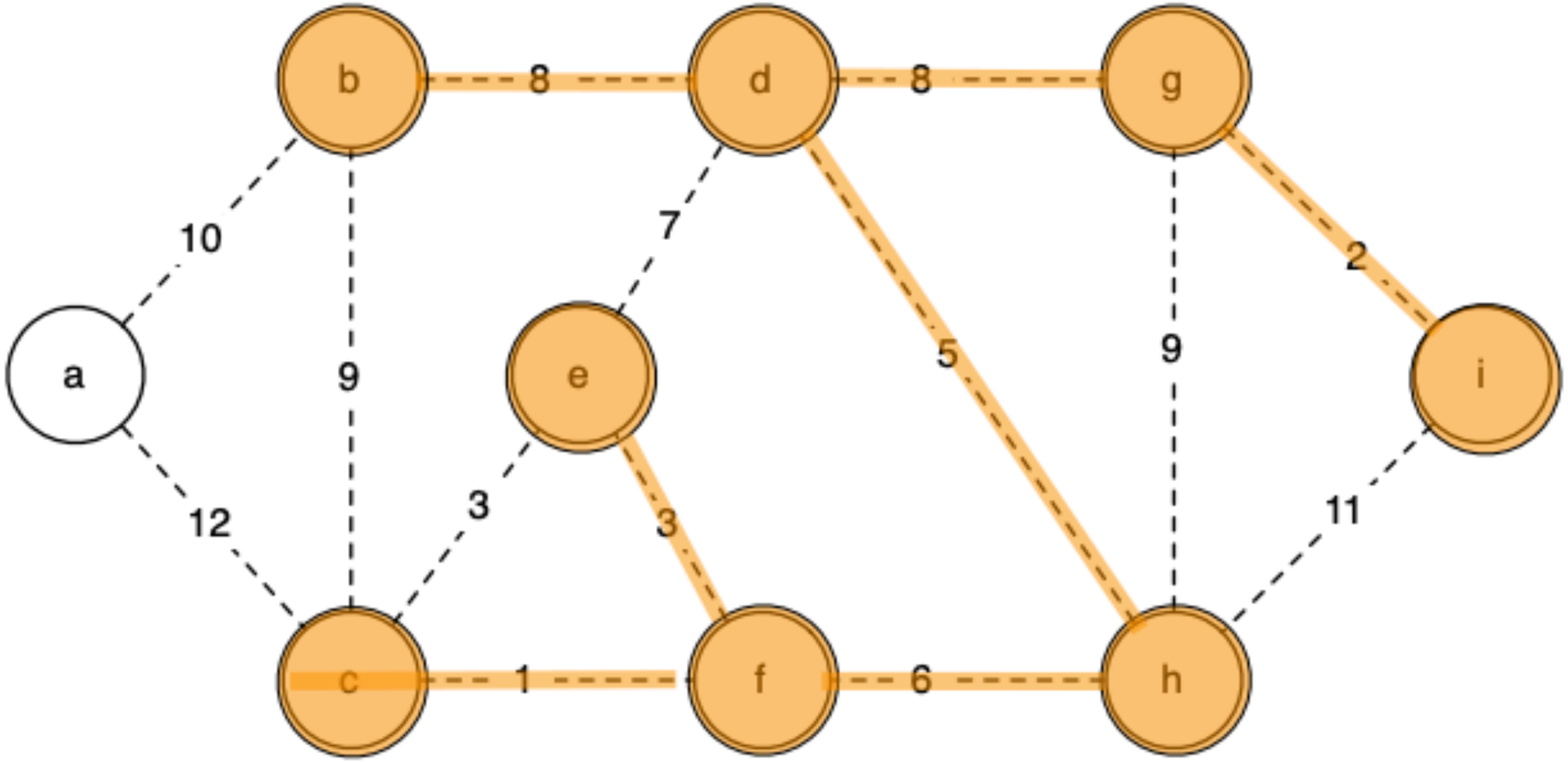
Example



Example

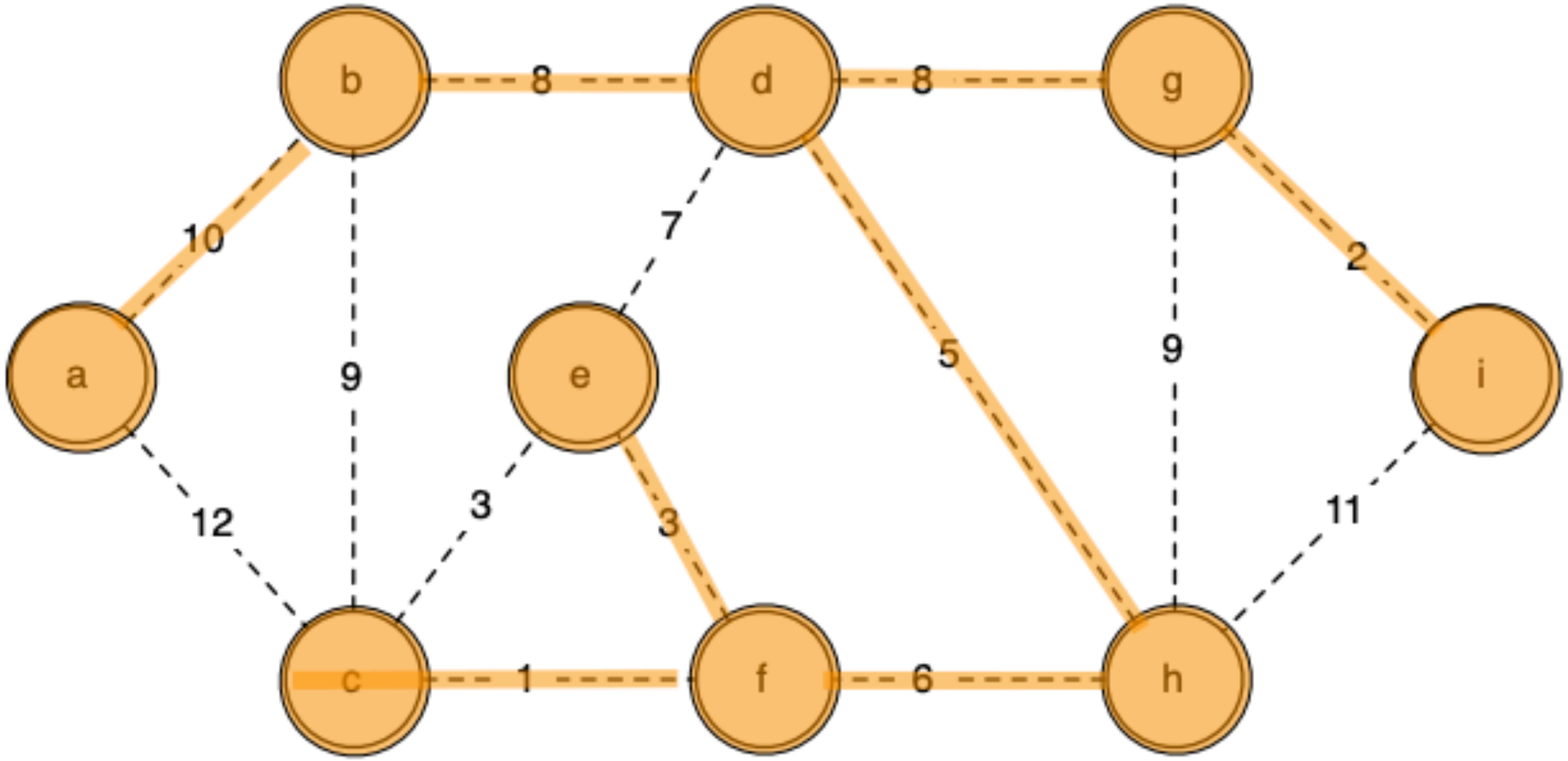


Example

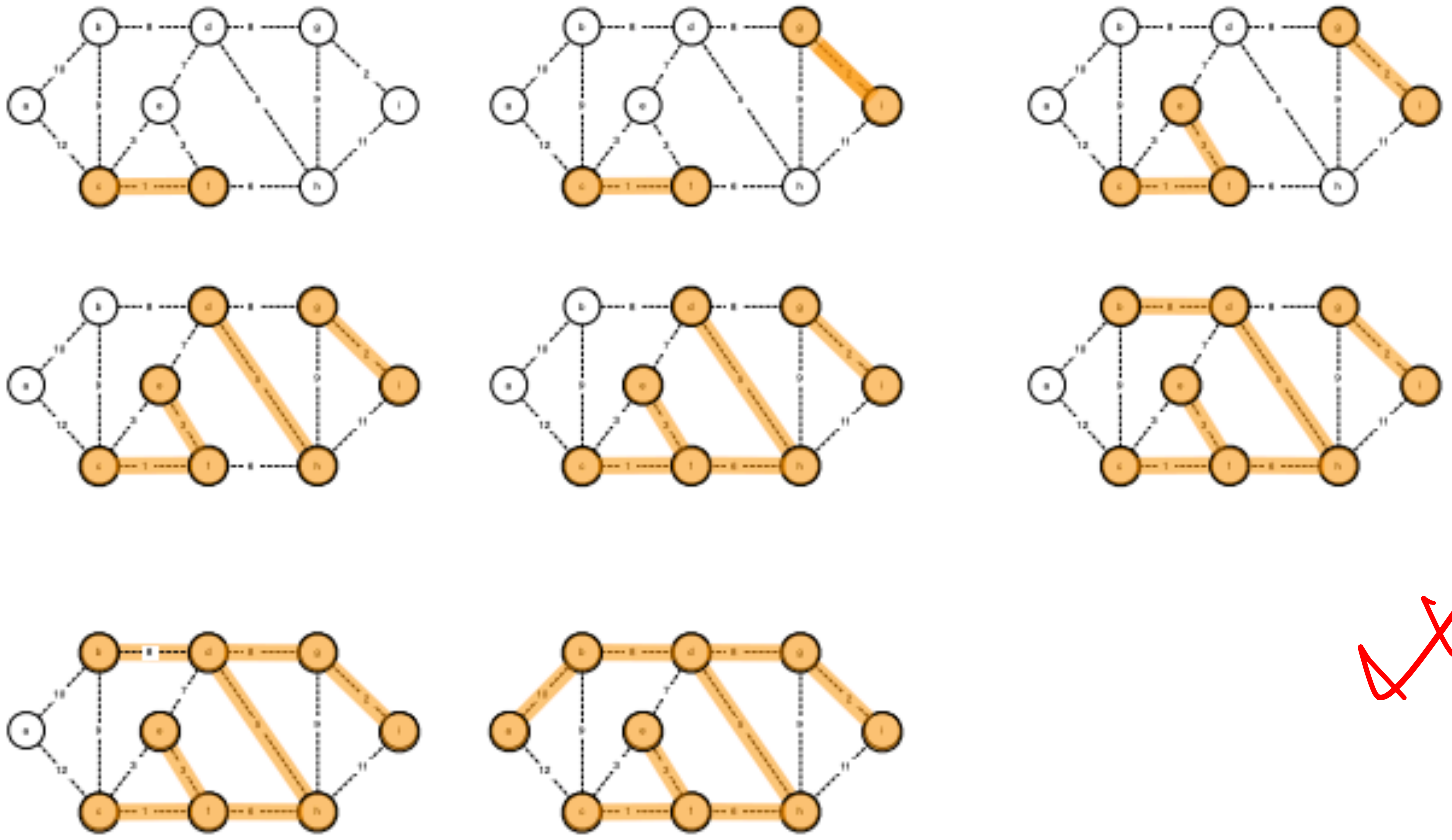


Example

MST



Kruskal's algorithm



Why does this work?

$T \leftarrow \emptyset$

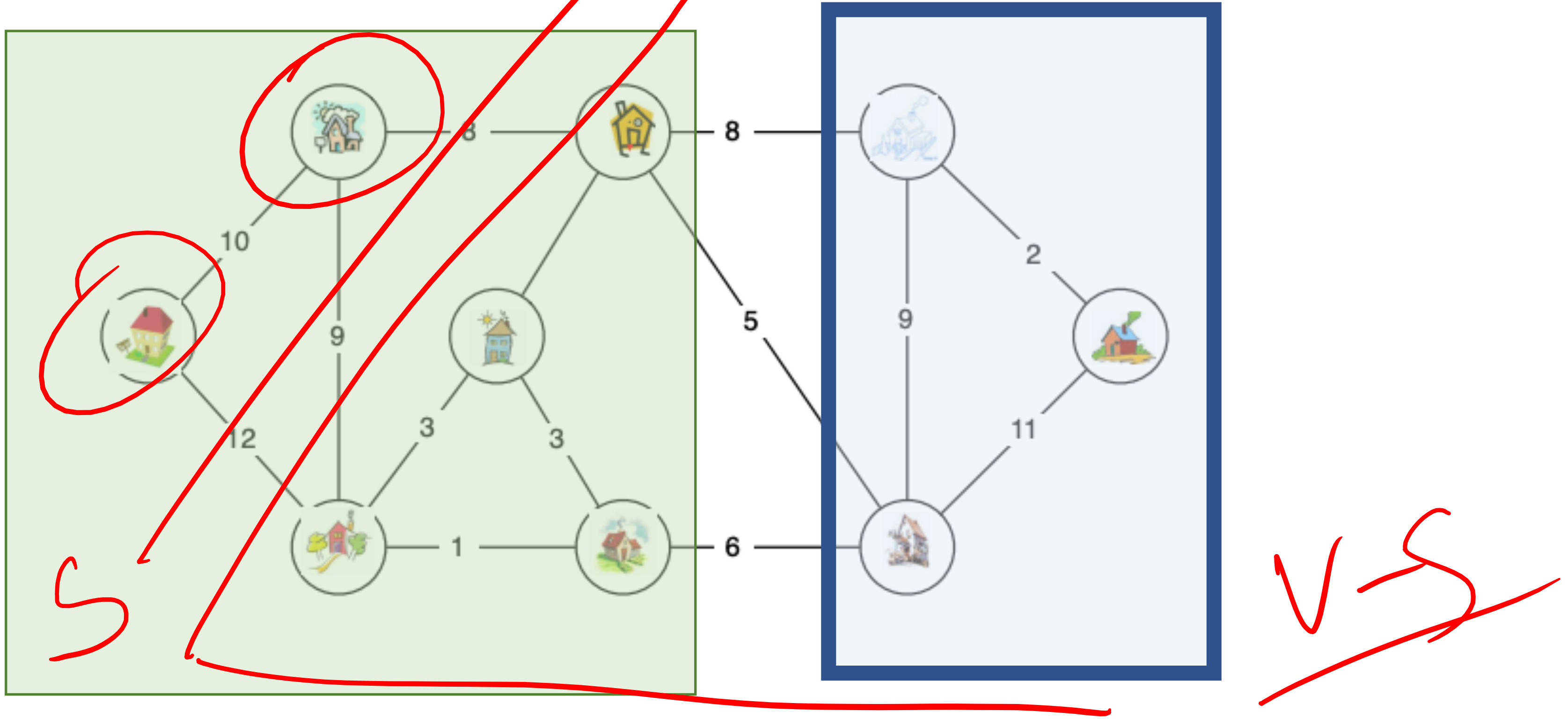
repeat $V - 1$ times:

add to T the lightest edge $e \in E$ that does not create a cycle

Definition: CUT

$(V) = S \cup (V-S)$

- CUT: a cut of G=(V,E) is a partition of V into 2 sets (S, V-S)

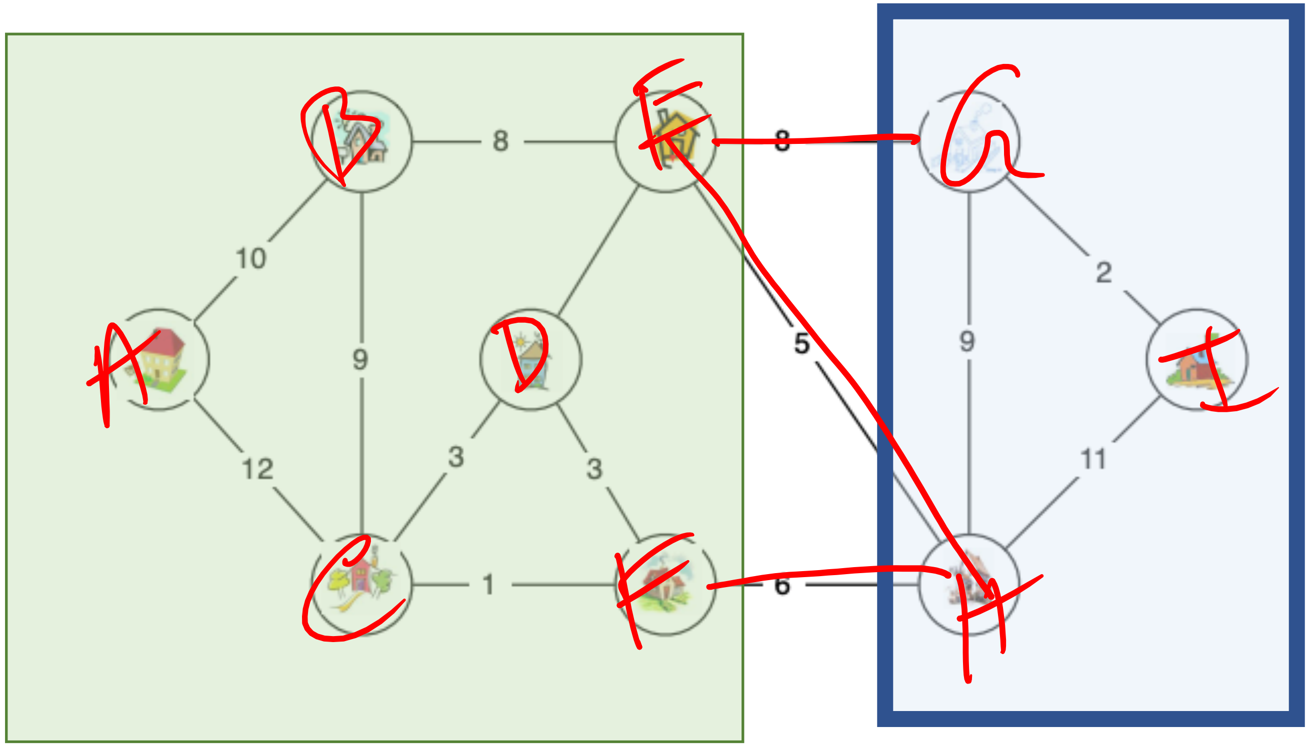


Definition: Crossing a CUT

$e_1 = E,G$
$e_2 = E,H$
$ce = F,H$

- An edge **e=(u,v)** crosses a cut (S,V-S) if **u ∈ S**, and **u ∈ V-S**

S



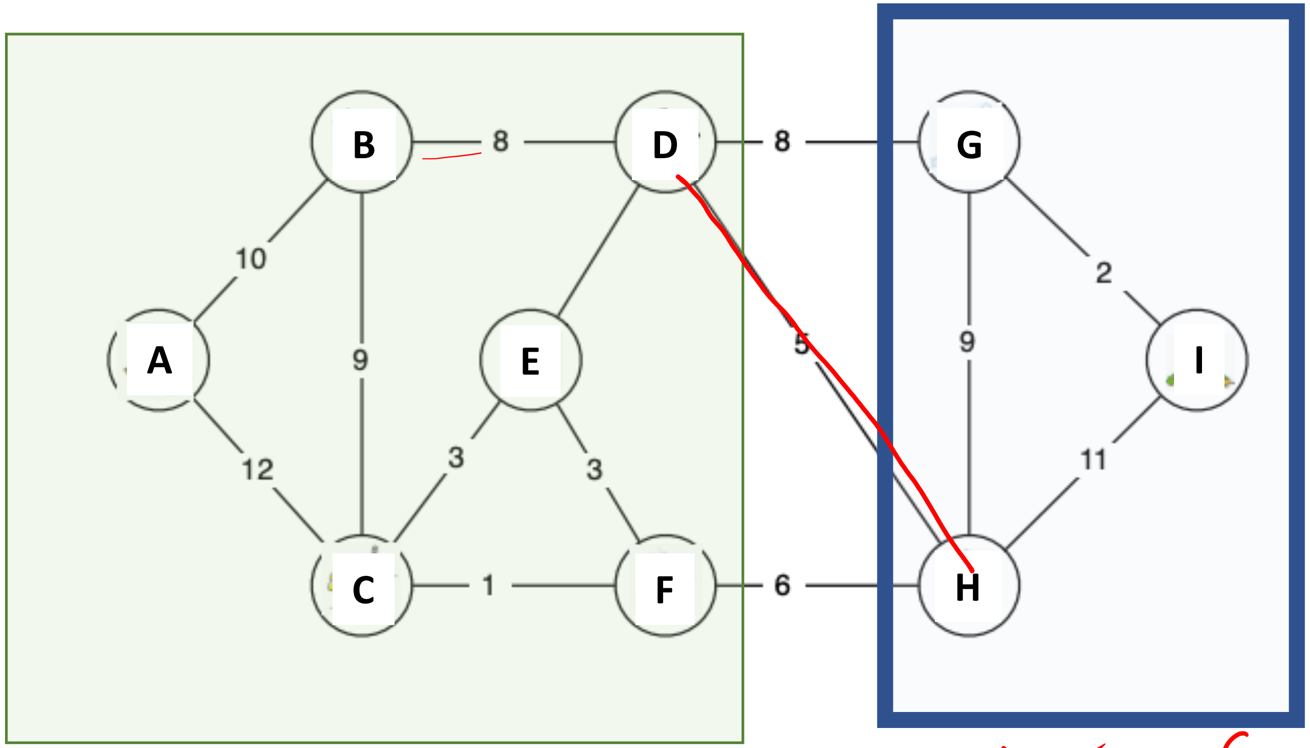
V-S

Definition: Respect

respects = E − {cutting cross}

- A set A respects the cut (S, V-S) if no edge e **∈ A, crosses (S,V-S)**

List of A



A, B
A, E
C, D
C, E
E, F
E, F
C, G
H I
G + I

Cut theorem

$A \subseteq MST$

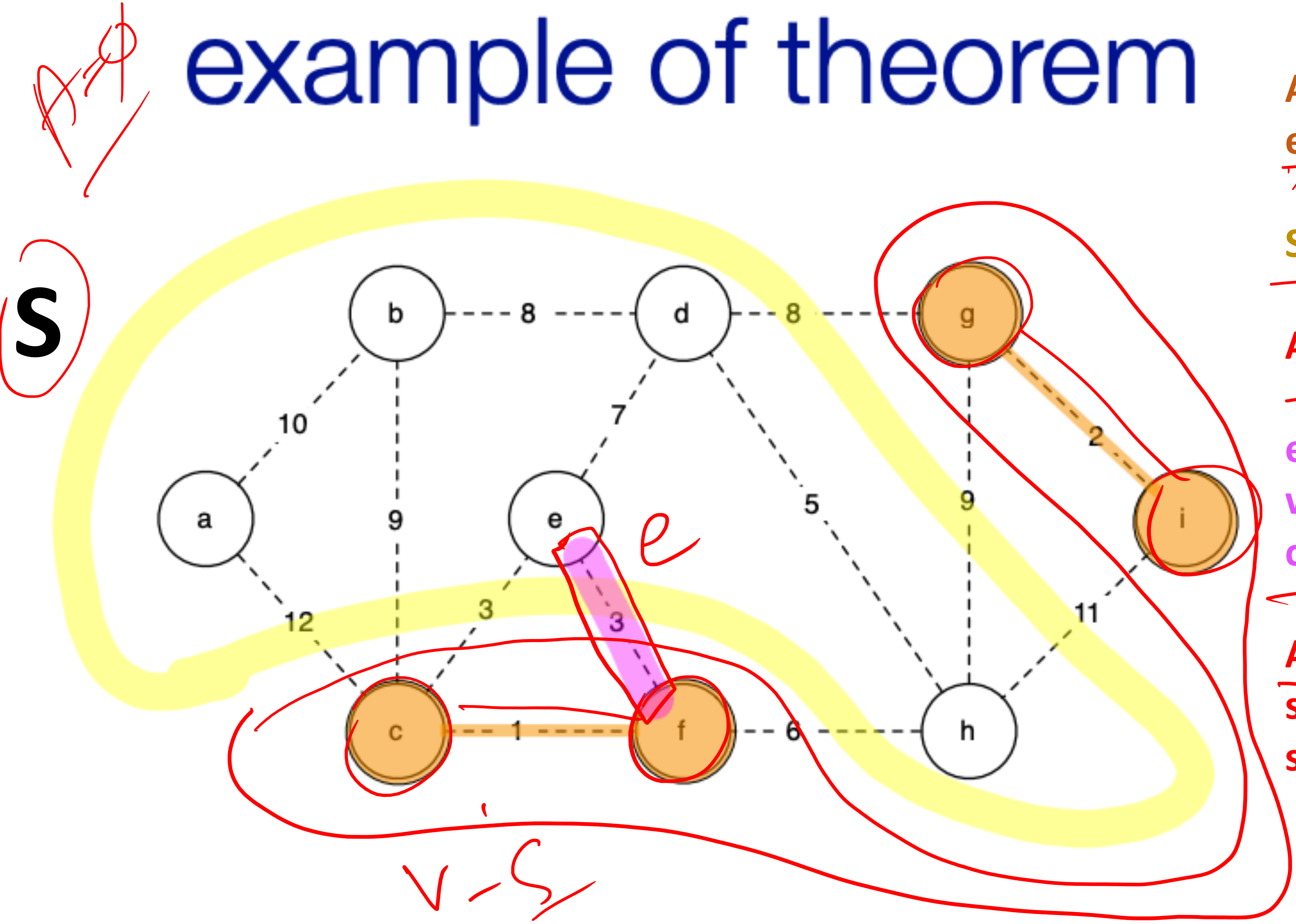
Suppose the set of edges A is part of an m.s.t.

Let $(S, V - S)$ be any cut that respects A.

Let edge e be the min-weight edge across $(S, V - S)$

Then: $A \cup \{e\}$ is part of an m.s.t.

example of theorem



A is the set of orange edges.

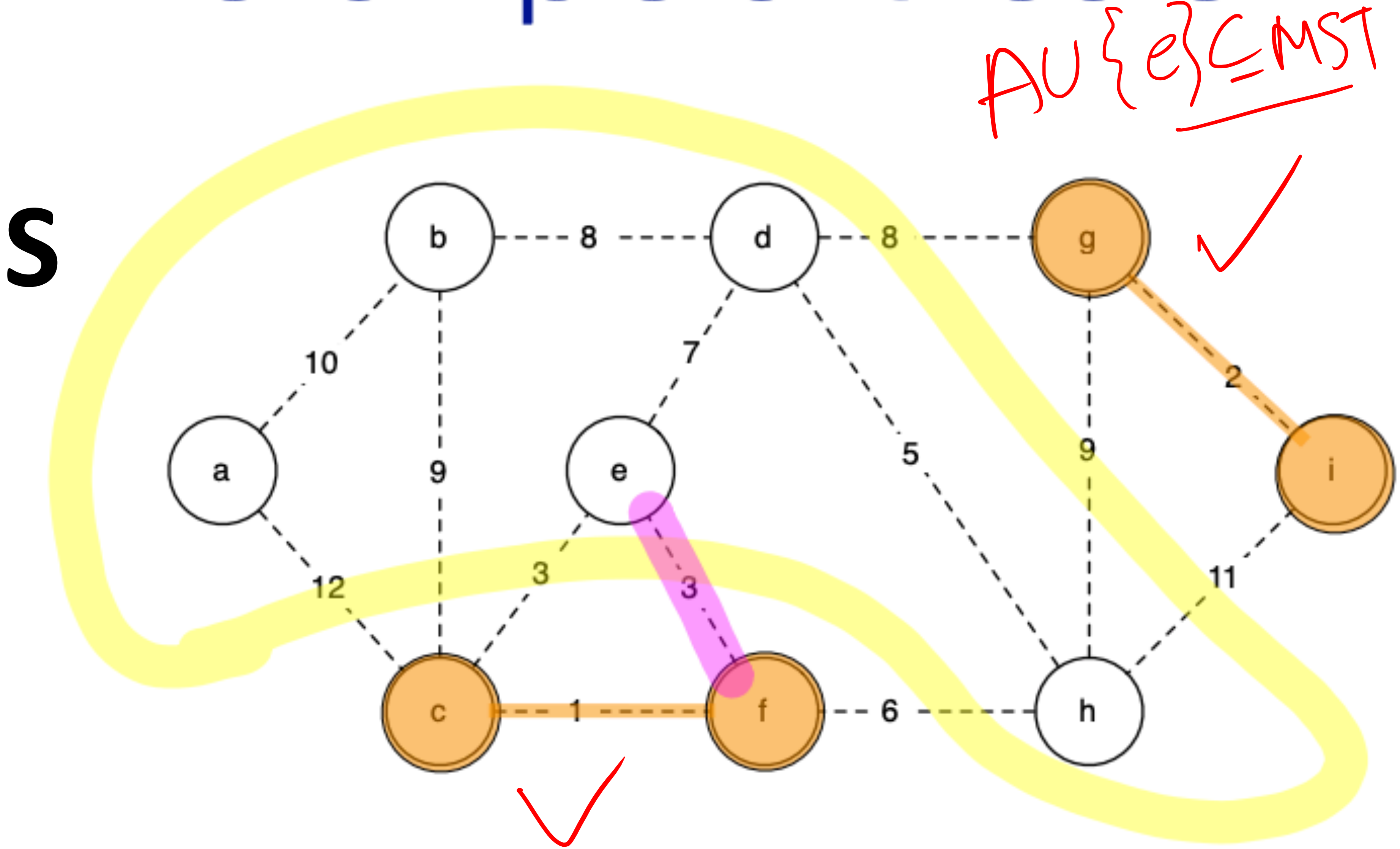
S is the cut.

A respects S.

e is the least weighted edge that crosses S.

A U {e} is part of some minimum spanning tree (MST)

example of theorem



A is the set of orange ~~edges.~~ (⊆ MST)

S is the cut.

A respects S.

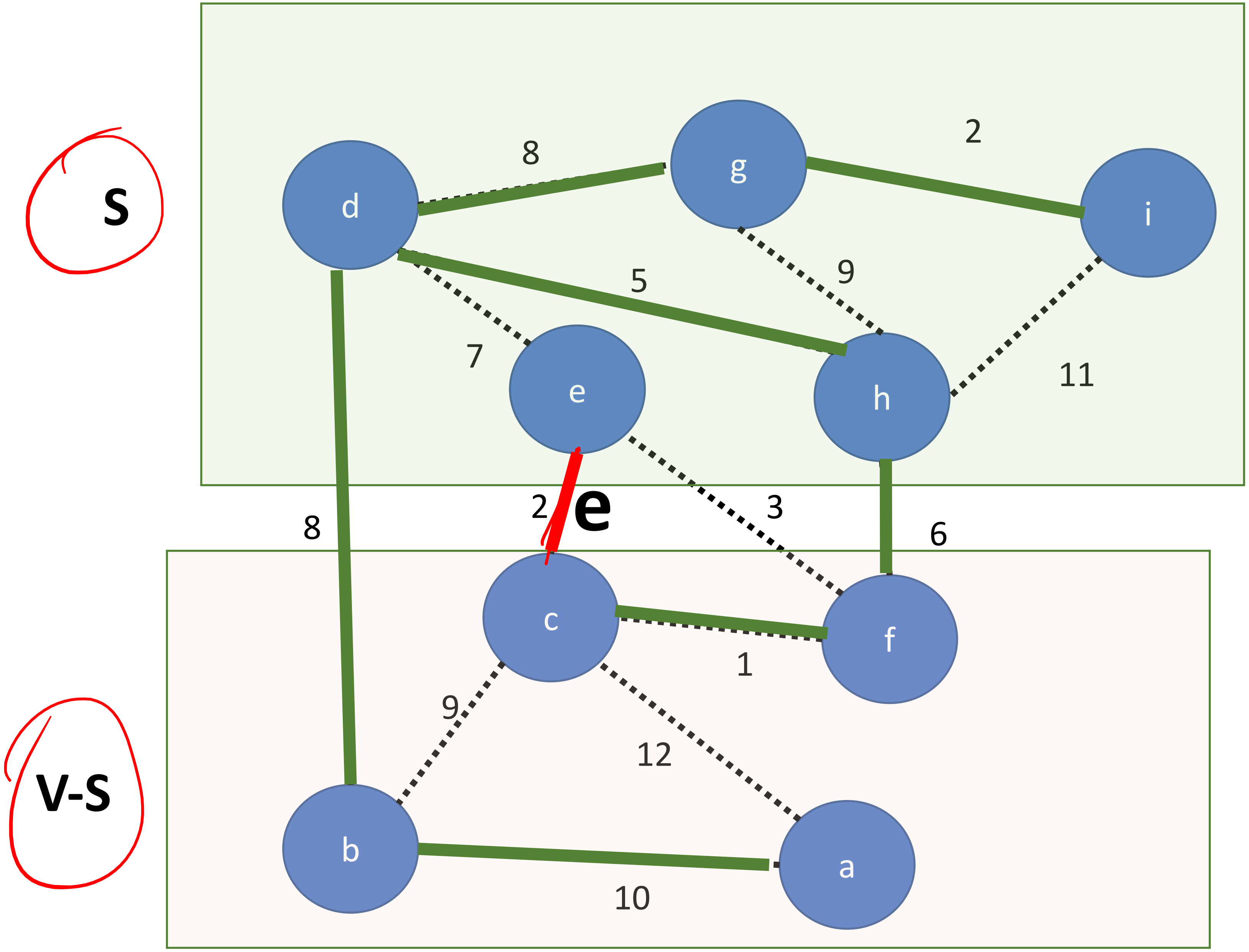
e is the least weighted edge that crosses S.

A ∪ {e} is part of some minimum spanning tree (MST)

Proof of cut theorem

Theorem: Suppose the set of edges A is part of a minimum spanning tree T if graph G=(V,E). Let (S,V-S) be any cut that respects A and let e be the edge with minimum weight that crosses (S, V-S). Then the set A U {e} is part of a minimum spanning tree.

Case 1: A U {e} is already part of T, theorem holds.


S
V-S
d
g
i
e
h
c
f
b
a
8
2
5
9
7
11
2
e
3
6
8
1
9
12
10

Proof of cut theorem

Theorem: Suppose the set of edges A is part of a minimum spanning tree T if graph G=(V,E). Let (S,V-S) be any cut that respects A and let e be the edge with minimum weight that crosses (S, V-S). Then the set A U {e} is part of a minimum spanning tree.

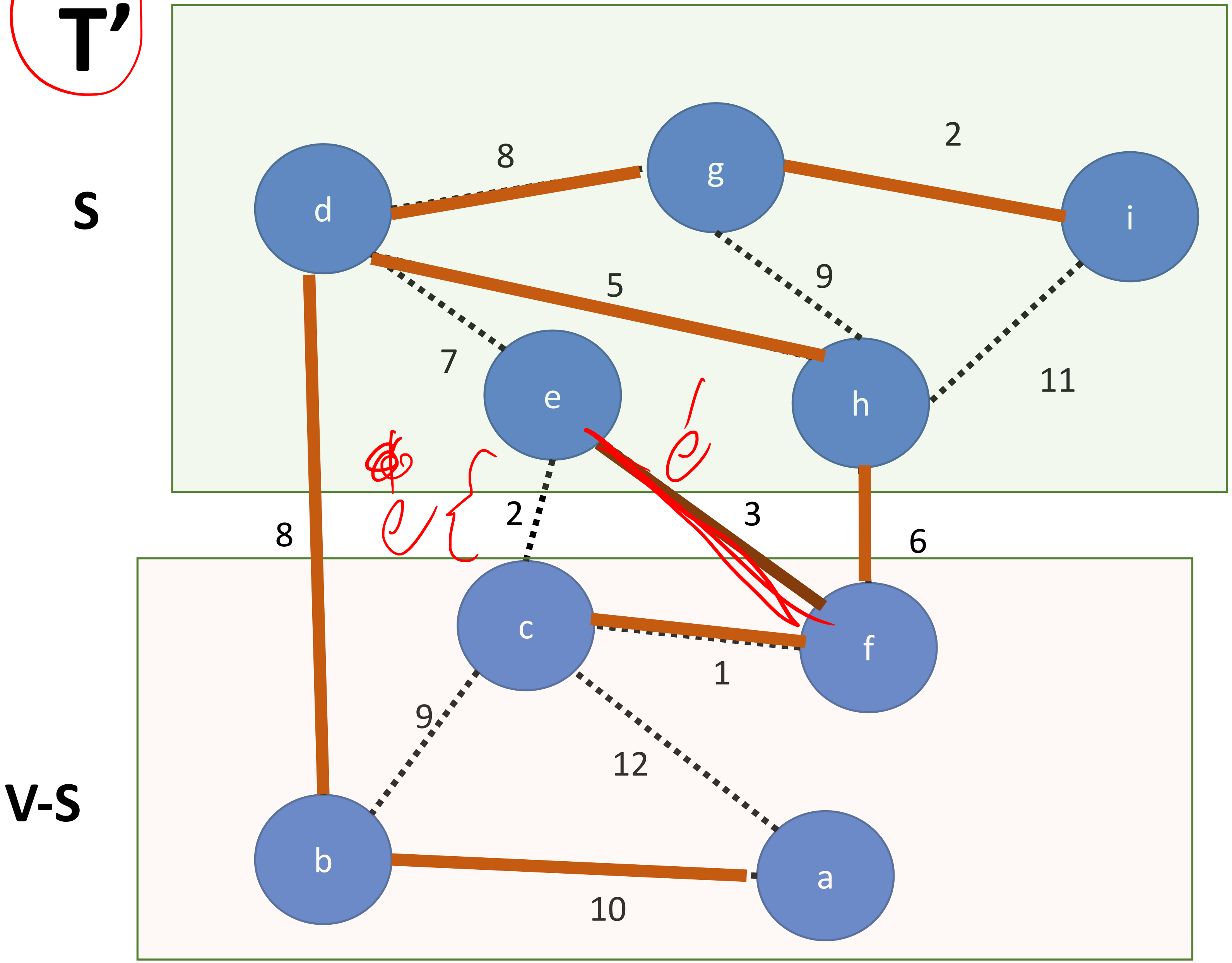
Case 1: A U {e} is already part of T, theorem holds.

Case 2: Otherwise. Minimum Spanning tree is T'.

$A \cup \{e\} \not\subseteq T$

$A \cup \{e\} \subseteq T \neq T'$

T’



Proof of cut theorem

A ∪ {e} ⊆ MST

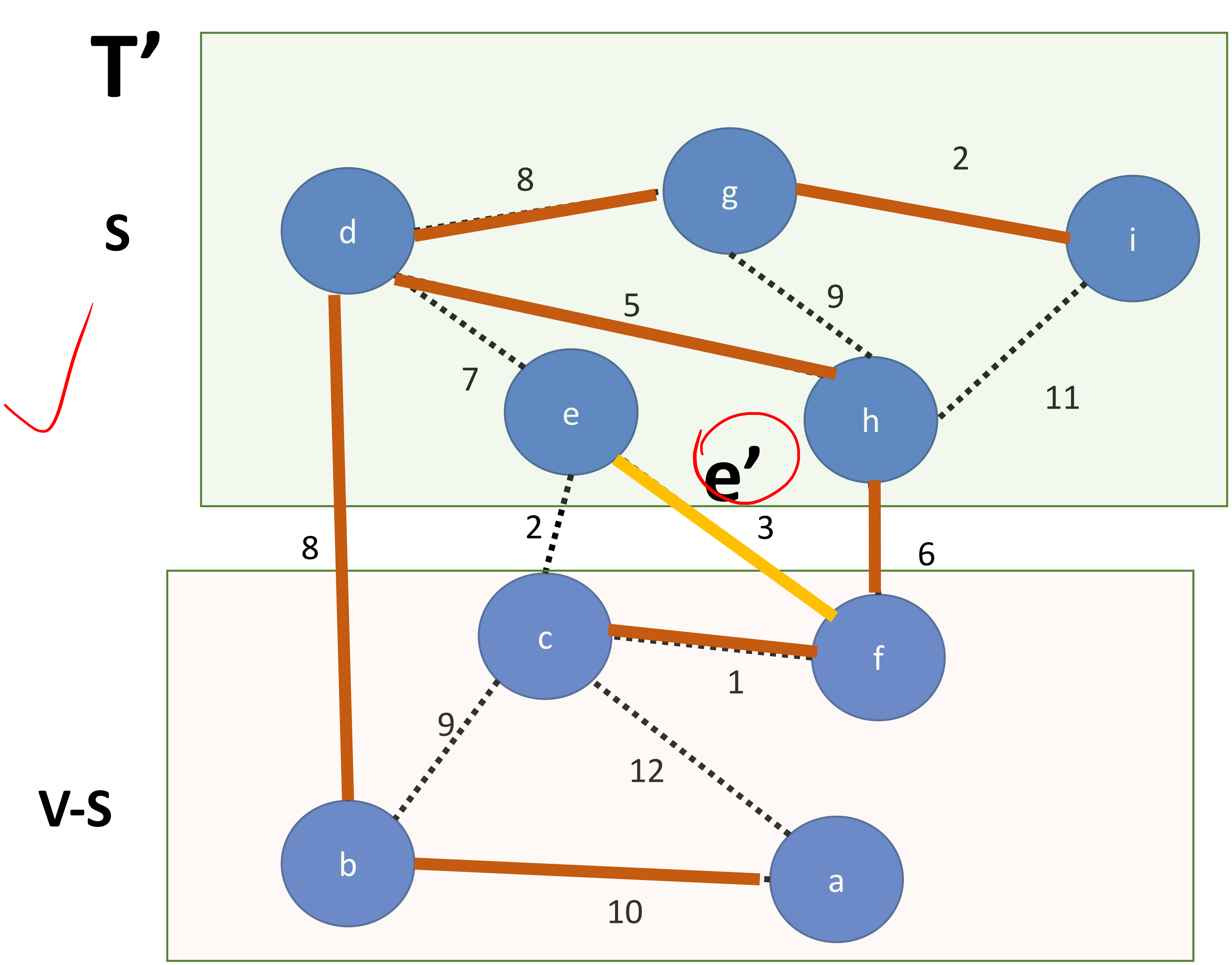
Theorem: Suppose the set of edges A is part of a minimum spanning tree T if graph G=(V,E). Let (S,V-S) be any cut that respects A and let e be the edge with minimum weight that crosses (S, V-S). Then the set A U {e} is part of a minimum spanning tree.

Case 1: A U {e} is already part of T, theorem holds.

Case 2: Otherwise. Minimum Spanning tree is T'.

Since, T' is an MST, there must be already an path from u to v

Let e' is the least weighted edge of that path (from u to v) that crossed the cut (S, V-S)



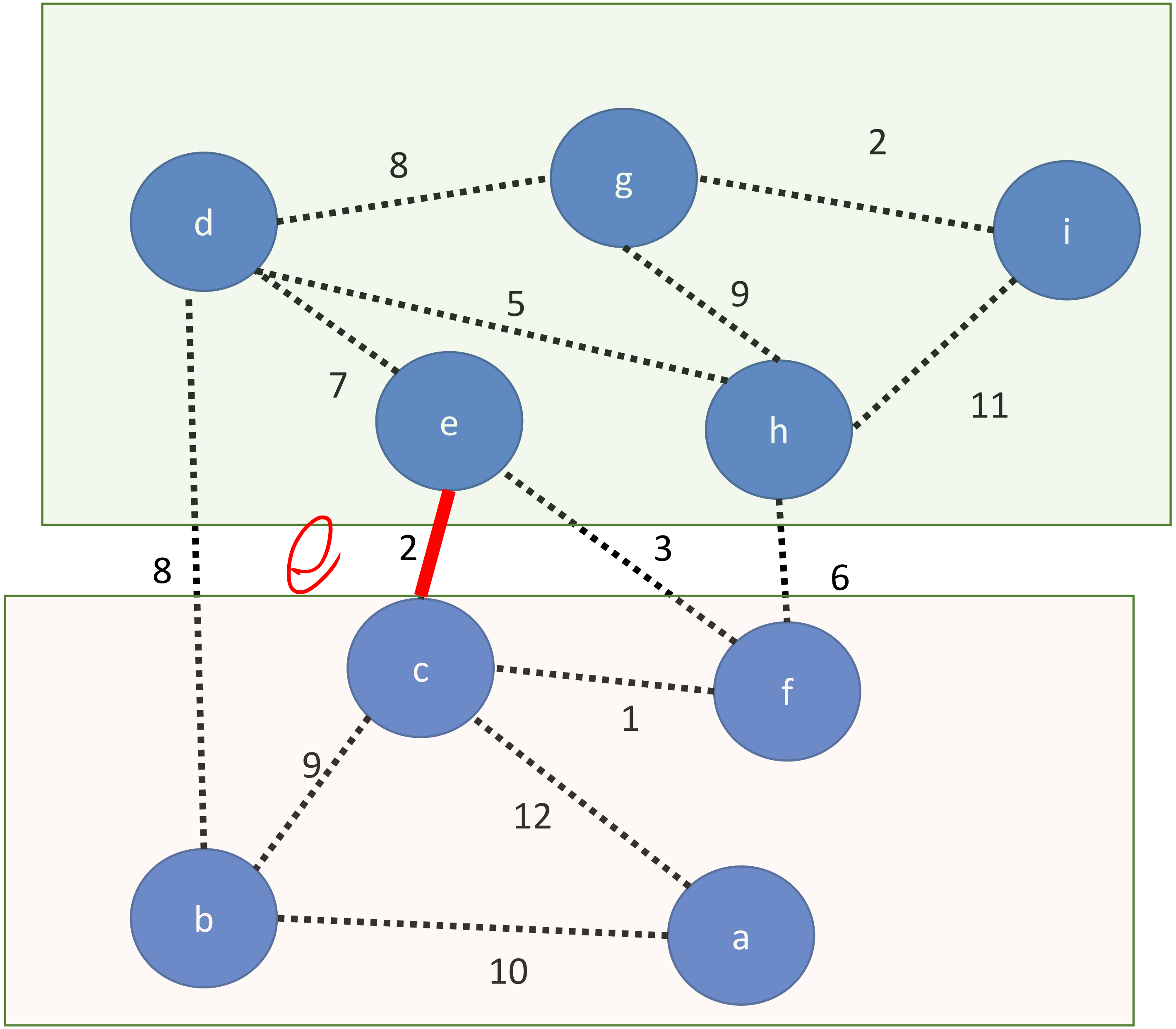
① $\in T'$

② crossing cut

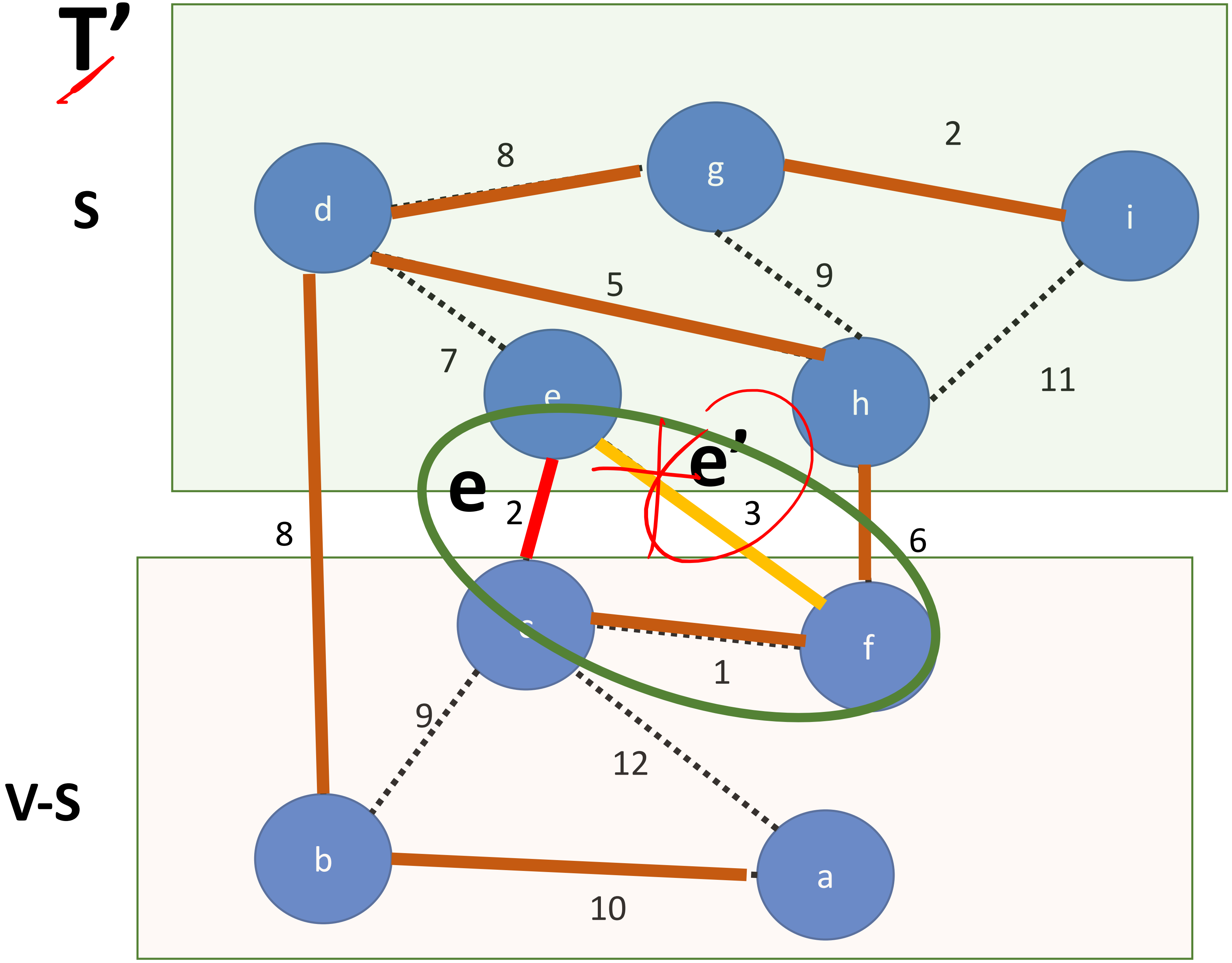
$\{ec\}$

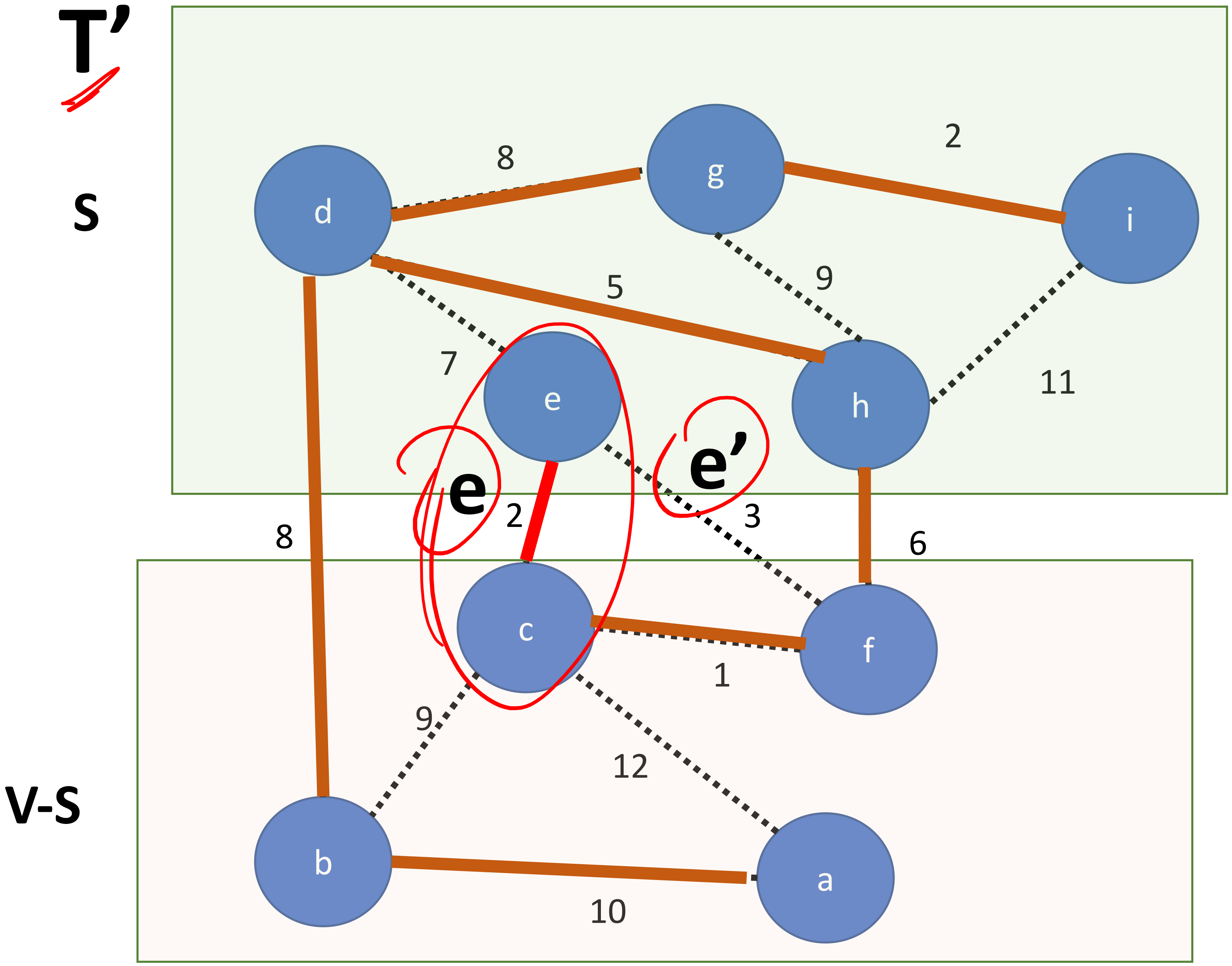
S

V-S



T'





$w(e') > w(e)$

$T = T' - w(e') + w(e)$

$\boxed{T < T'}$

T' MST

Contradiction!

T' was MST!

correctness

```
KRUSKAL-PSEUDOCODE(G)
1  A ← ∅
2  repeat  V − 1 times:
3          add to A the lightest edge e ∈ E that does not create a cycle
```

Proof: by induction. in step 1, A is part of some MST.
Suppose that after k steps, A is part of some MST (line 2).
In line 3, we add an edge e=(u,v).

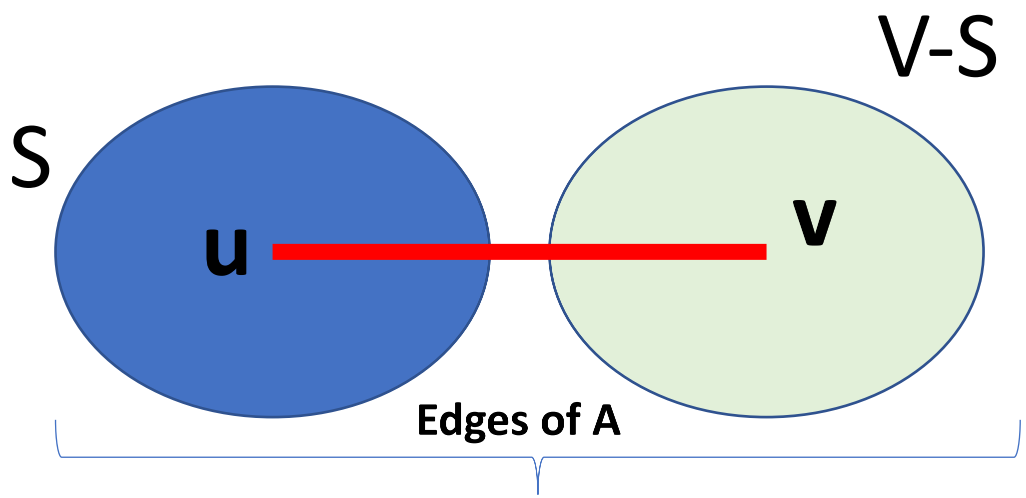
correctness

```
KRUSKAL-PSEUDOCODE(G)
1  A ← ∅
2  repeat  V − 1 times:
3              add to A the lightest edge e ∈ E that does not create a cycle
```

Proof: by induction. in step 1, A is part of some MST. Suppose that after k steps, A is part of some MST (line 2). In line 3, we add an edge e=(u,v).

A respect the cut

CASE 1:



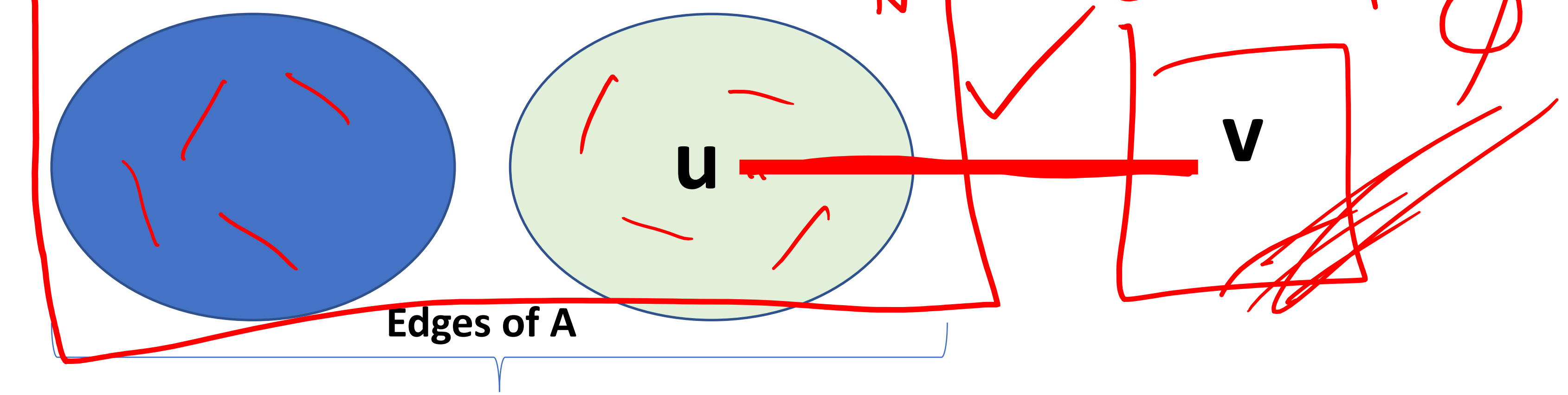
correctness

KRUSKAL-PSEUDOCODE(G)

```
1  A ← ∅
2  repeat  V − 1 times:
3          add to A the lightest edge e ∈ E that does not create a cycle
```

Proof: by induction. in step 1, A is part of some MST.
Suppose that after k steps, A is part of some MST (line 2).
In line 3, we add an edge e=(u,v).

CASE 2:

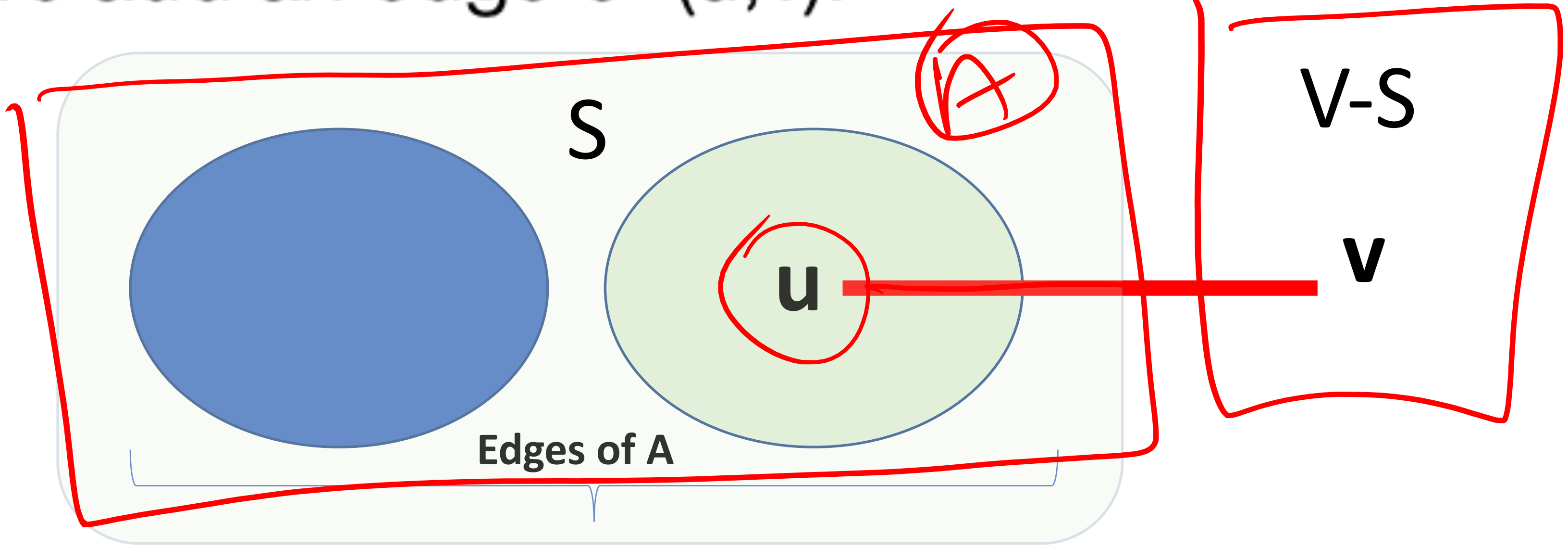


correctness

```
KRUSKAL-PSEUDOCODE(G)
1  A ← ∅
2  repeat V − 1 times:
3          add to A the lightest edge e ∈ E that does not create a cycle
```

Proof: by induction. in step 1, A is part of some MST. Suppose that after k steps, A is part of some MST (line 2). In line 3, we add an edge e=(u,v).

CASE 2:



$A = \emptyset$

correctness

KRUSKAL-PSEUDOCODE(G)

1 $A \leftarrow \emptyset$

2 **repeat** $V - 1$ times:

3 add to A the lightest edge $e \in E$ that does not create a cycle

$V - S$

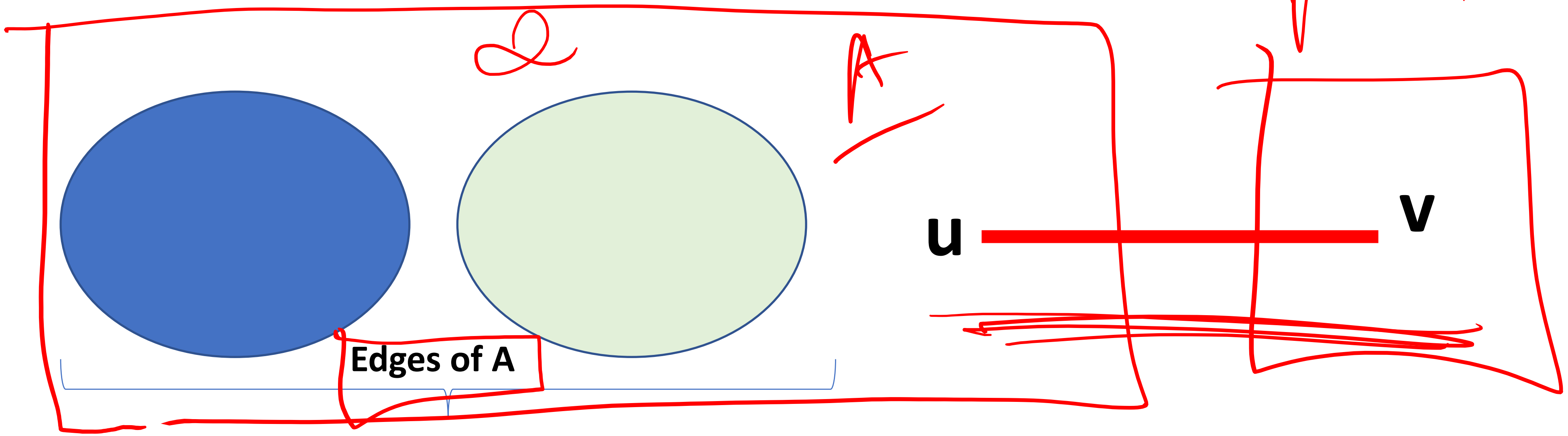
Proof: by induction. in step 1, A is part of some MST. Suppose that after k steps, A is part of some MST (line 2). In line 3, we add an edge e=(u,v).

S

$A = \emptyset$

CASE 3:

P

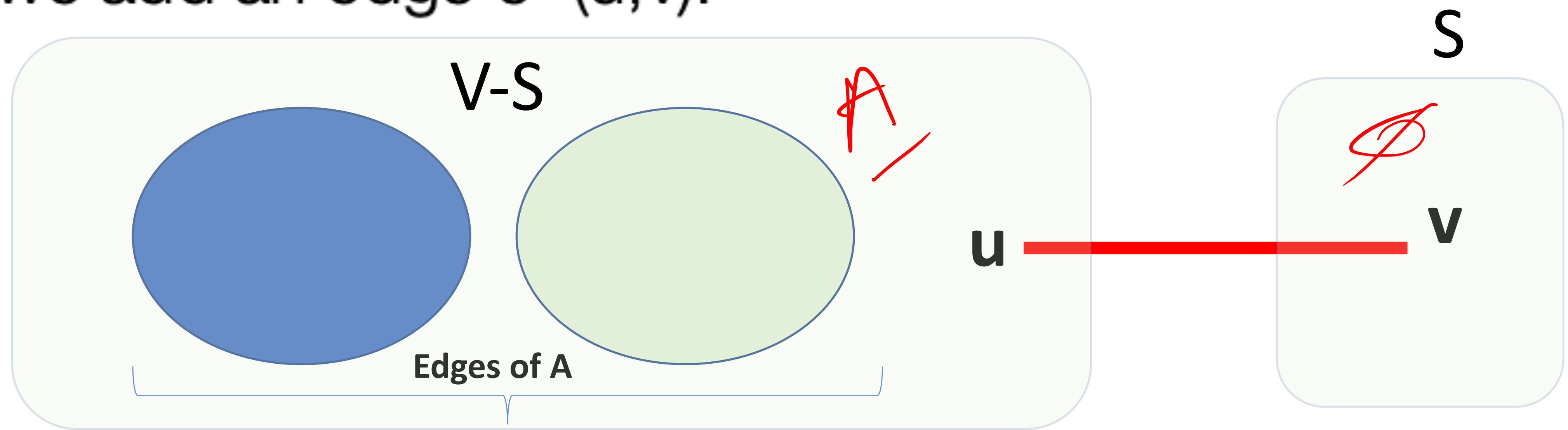


correctness

```
KRUSKAL-PSEUDOCODE(G)
1  A ← ∅
2  repeat  V − 1 times:
3          add to A the lightest edge e ∈ E that does not create a cycle
```

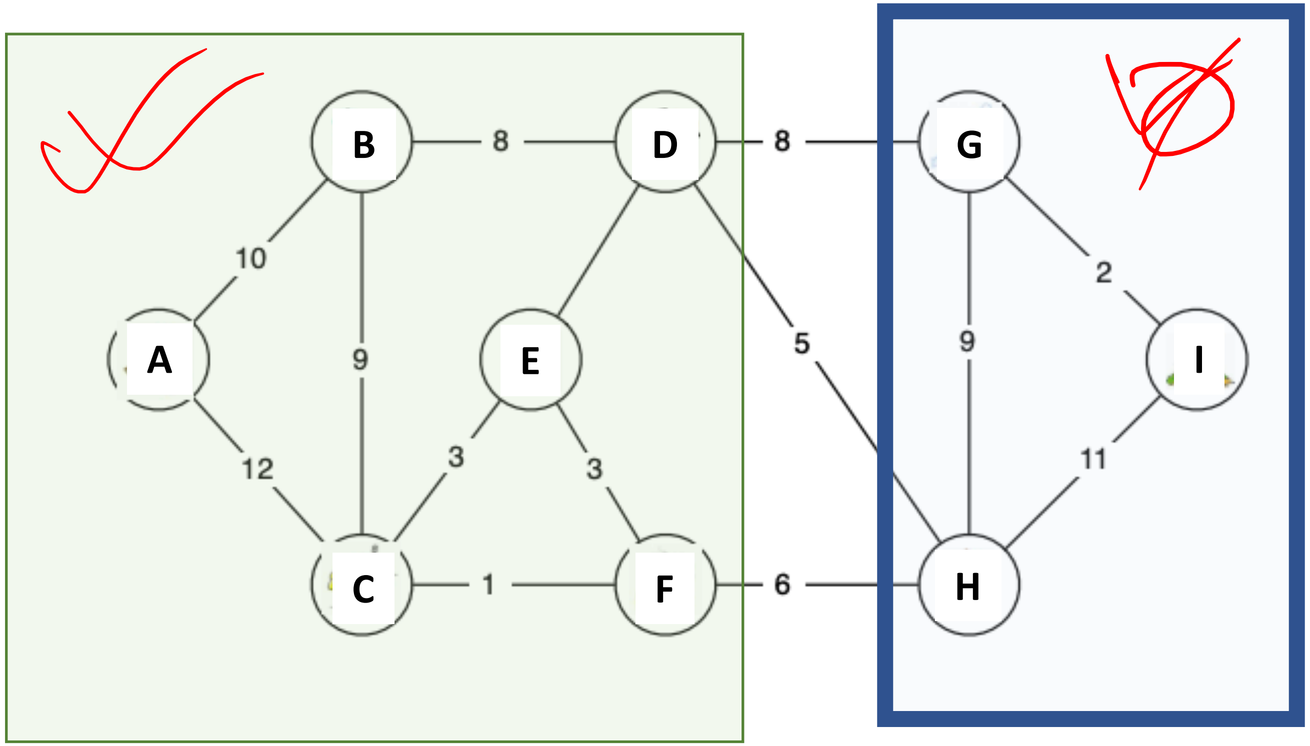
Proof: by induction. in step 1, A is part of some MST. Suppose that after k steps, A is part of some MST (line 2). In line 3, we add an edge e=(u,v).

CASE 3:



Definition: Respect

- A set A respects the cut (S, V-S) if no edge e **∈ A, crosses (S,V-S)**



List of A

Cut theorem

Suppose the set of edges A is part of an m.s.t.

Let $(S, V - S)$ be any cut that respects A.

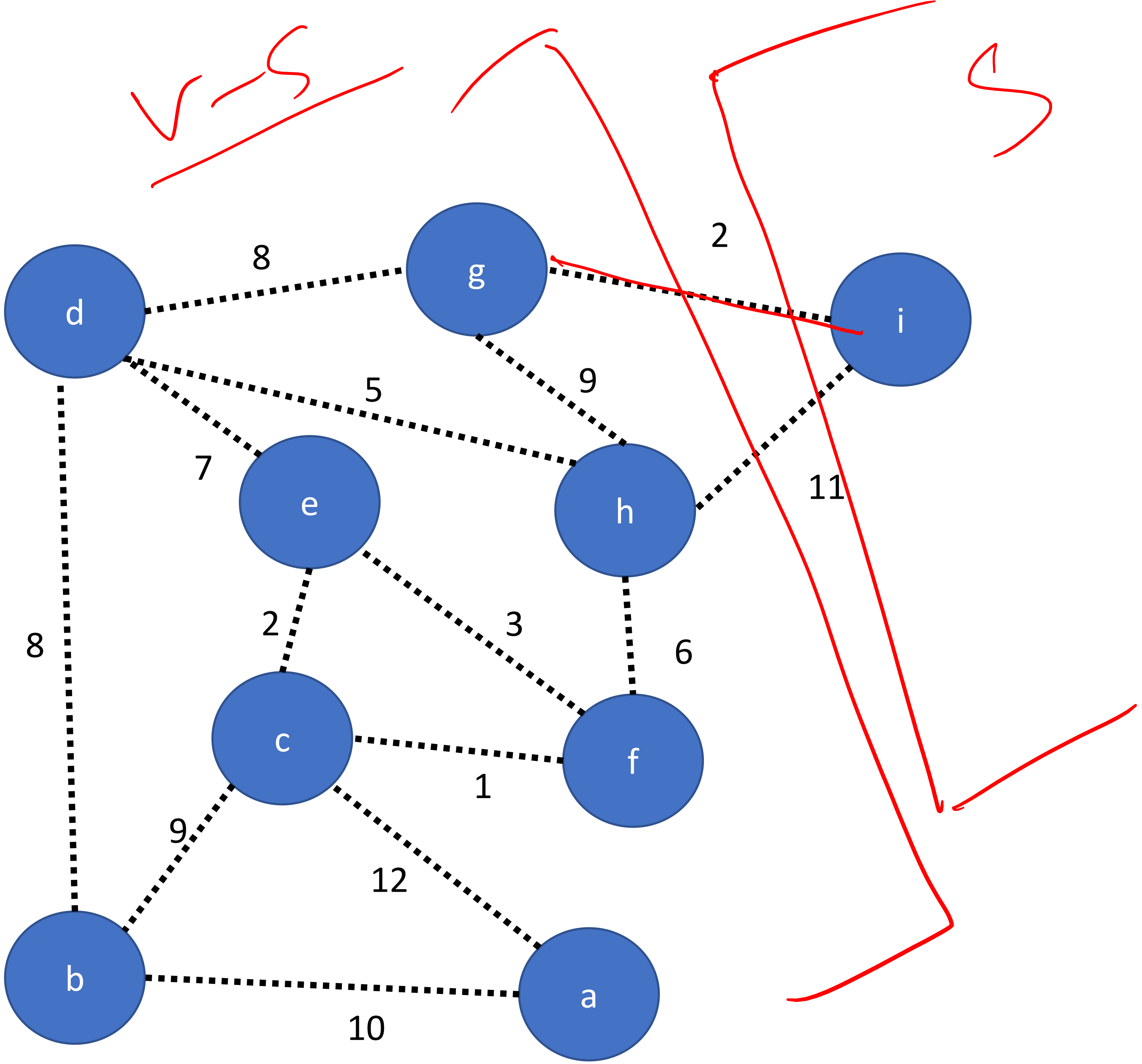
Let edge e be the min-weight edge across $(S, V - S)$

Then: $A \cup \{e\}$ is part of an m.s.t.

```
GENERAL-MST-STRATEGY(G = (V, E))
A ← ∅
repeat V − 1 times:
    Pick a cut (S, V − S) that respects A
    Let e be min-weight edge over cut (S, V − S)
    A ← A ∪ {e}
```

Algorithm

A= Ø

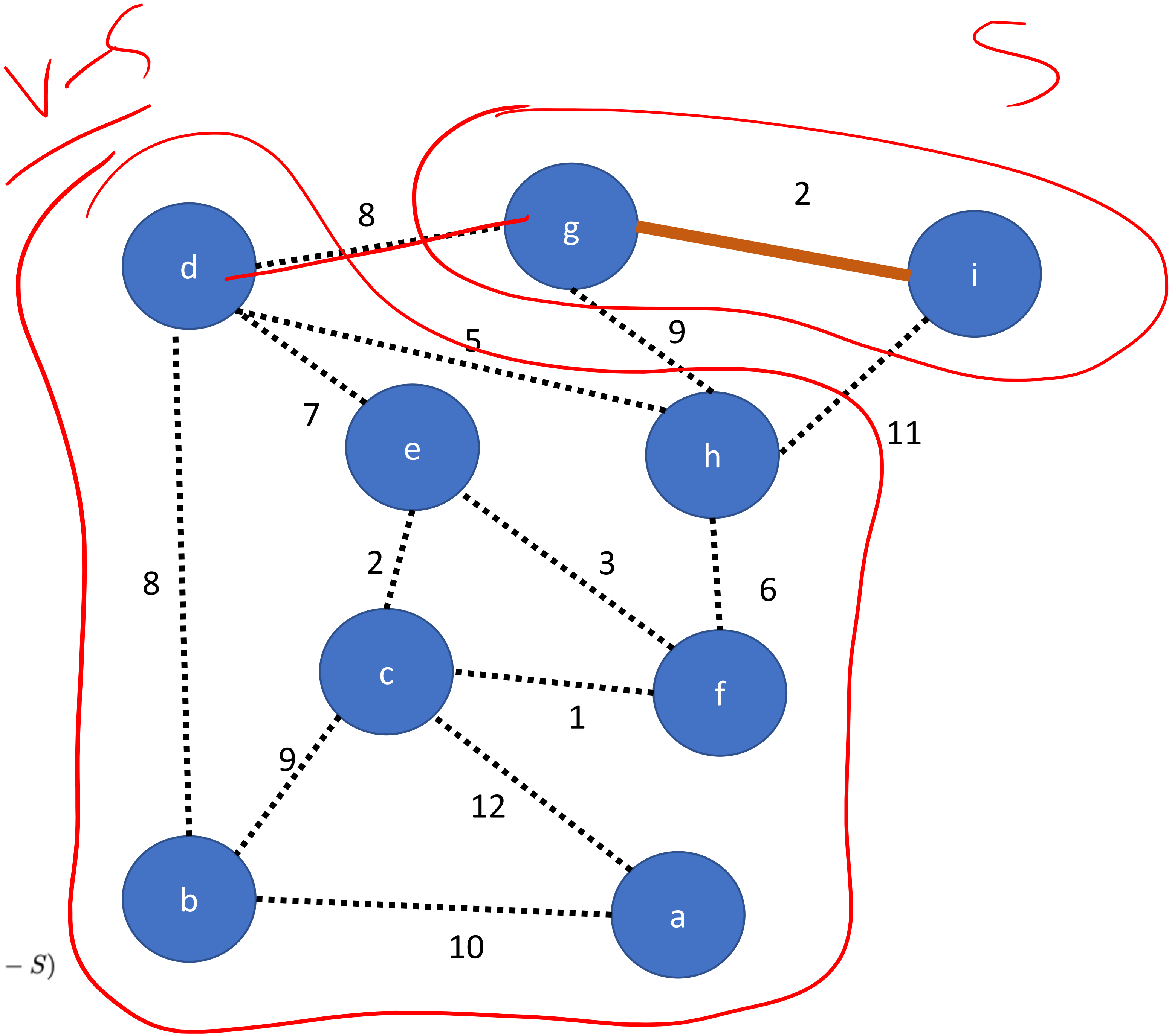


GENERAL-MST-STRATEGY($G = (V, E)$)

```
A ← ∅
repeat V − 1 times:
    Pick a cut (S, V − S) that respects A
    Let e be min-weight edge over cut (S, V − S)
    A ← A ∪ {e}
```

Algorithm

A= {(g,i)}

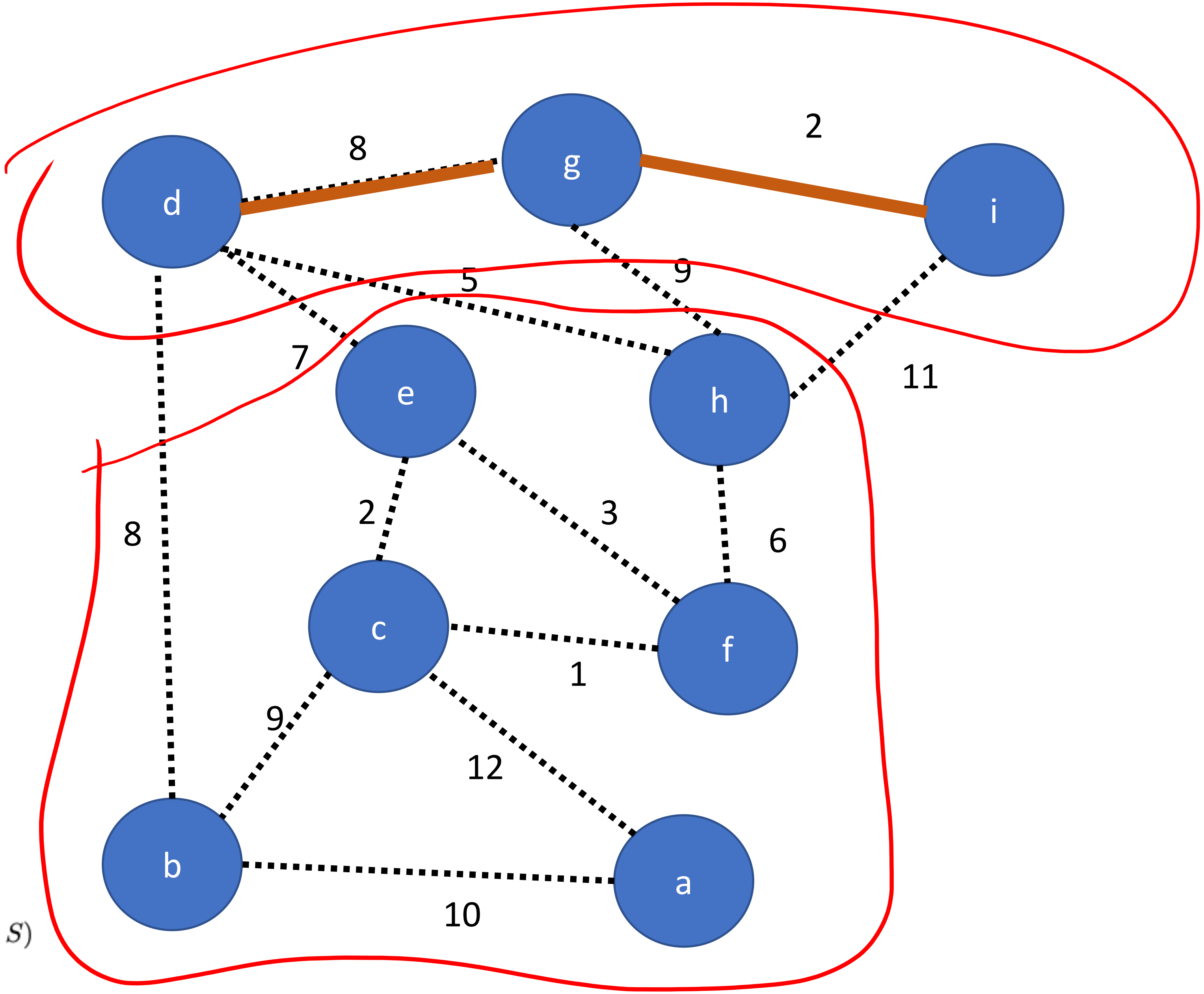


GENERAL-MST-STRATEGY$(G = (V, E))$

$A \leftarrow \emptyset$
repeat $V - 1$ times:
    Pick a cut $(S, V - S)$ that respects A
    Let e be min-weight edge over cut $(S, V - S)$
    $A \leftarrow A \cup \{e\}$

Algorithm

A= {(g,i), (d,g)}

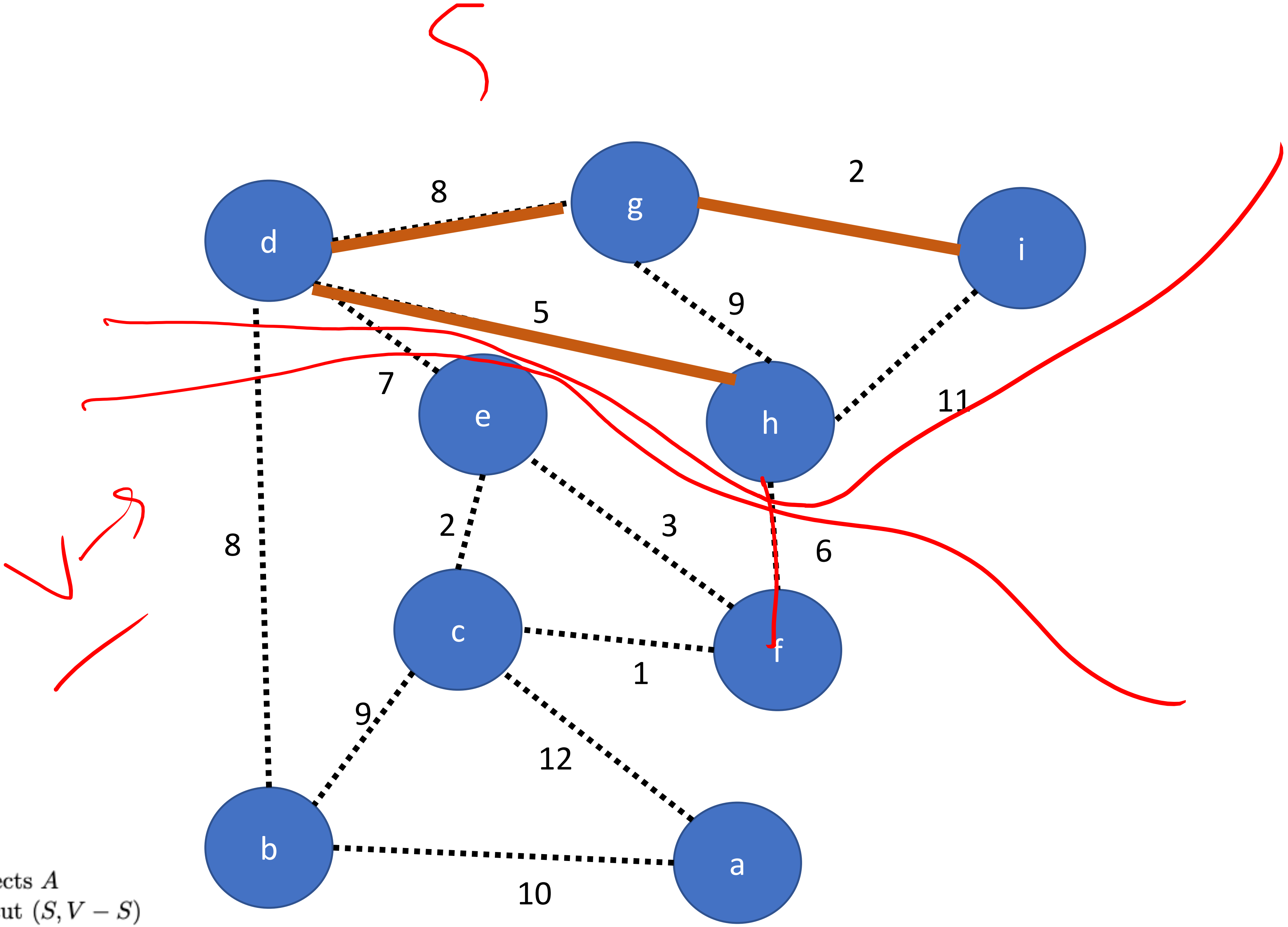


GENERAL-MST-STRATEGY$(G = (V, E))$

1 $A \leftarrow \emptyset$
2 **repeat** $V - 1$ times:
3 Pick a cut $(S, V - S)$ that respects A
4 Let e be min-weight edge over cut $(S, V - S)$
5 $A \leftarrow A \cup \{e\}$

Algorithm

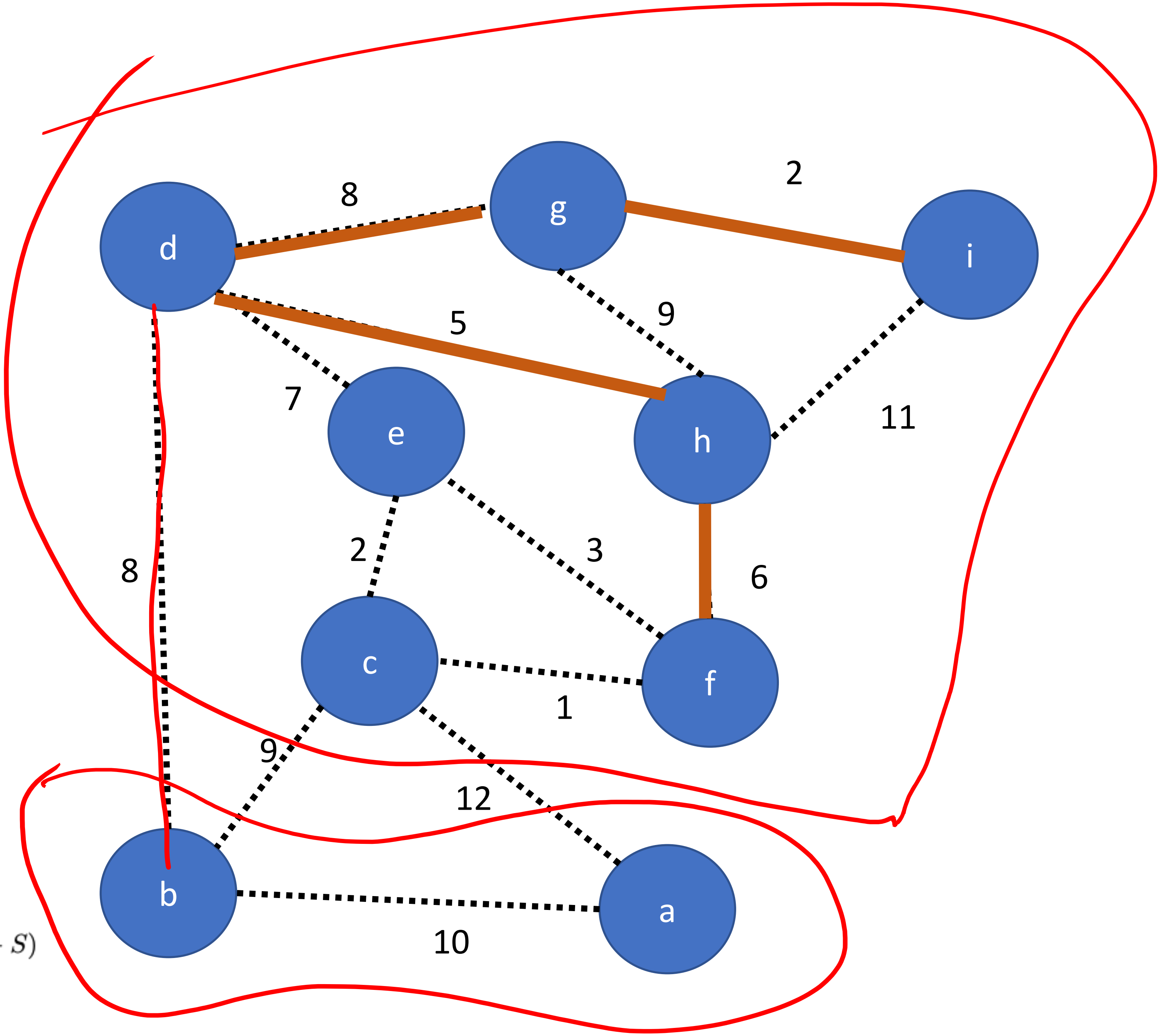
A= {(g,i), (d,g),(d,h)}



GENERAL-MST-STRATEGY($G = (V, E)$)
1 $A \leftarrow \emptyset$
2 **repeat** $V - 1$ times:
3 Pick a cut $(S, V - S)$ that respects A
4 Let e be min-weight edge over cut $(S, V - S)$
5 $A \leftarrow A \cup \{e\}$

Algorithm

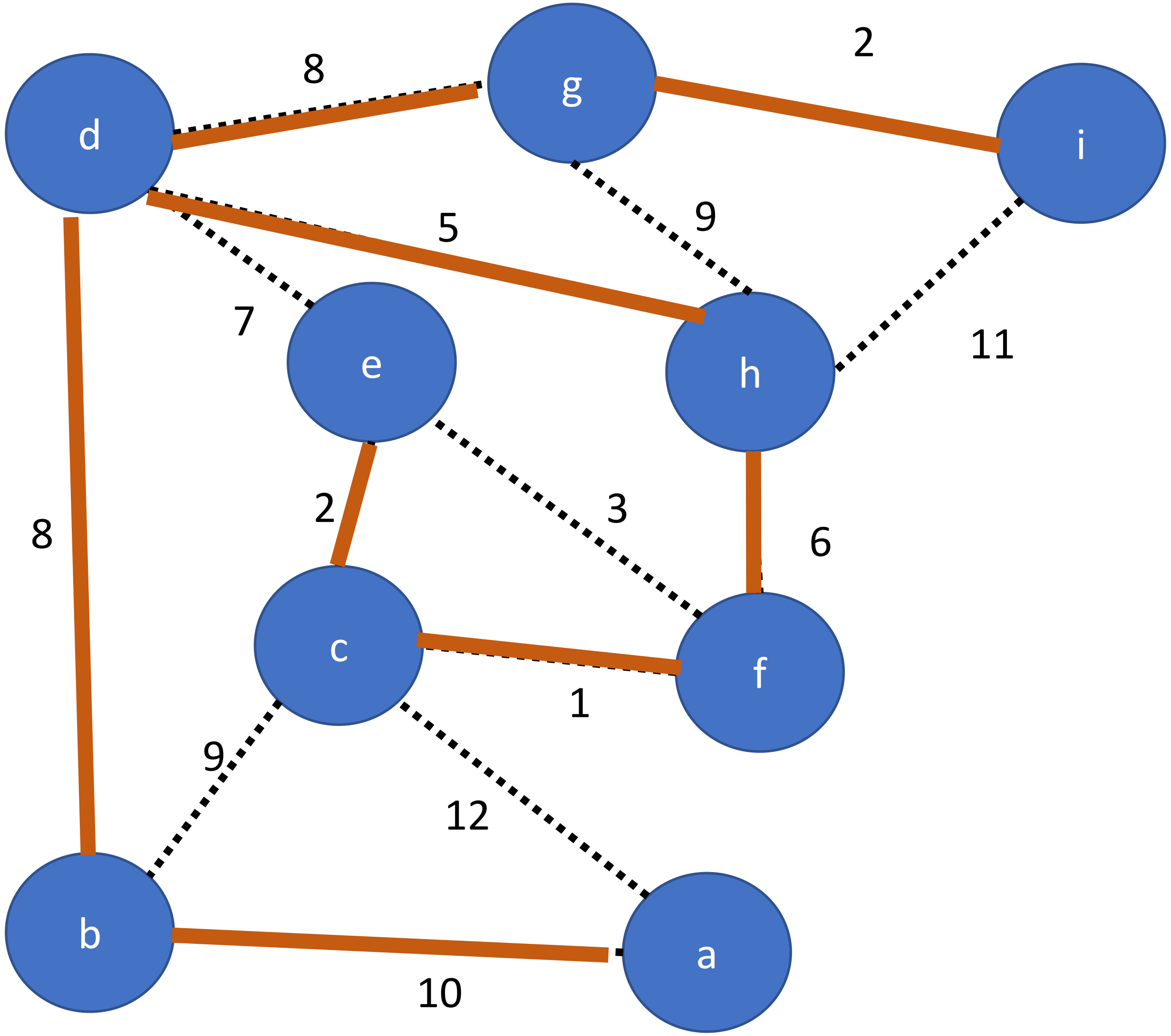
A= {(g,i), (d,g),(d,h), (h,f)}



GENERAL-MST-STRATEGY($G = (V, E)$)

1 $A \leftarrow \emptyset$
2 **repeat** $V - 1$ times:
3 Pick a cut $(S, V - S)$ that respects A
4 Let e be min-weight edge over cut $(S, V - S)$
5 $A \leftarrow A \cup \{e\}$

MST



GENERAL-MST-STRATEGY($G = (V, E)$)

$A \leftarrow \emptyset$
repeat $V - 1$ times:
    Pick a cut $(S, V - S)$ that respects A
    Let e be min-weight edge over cut $(S, V - S)$
    $A \leftarrow A \cup \{e\}$

Prim's Algorithm

General-MST-Strategy$(G = (V, E))$

$A \leftarrow \emptyset$
repeat $V - 1$ times:
    Pick a cut $(S, V - S)$ that respects A
    Let e be min-weight edge over cut $(S, V - S)$
    $A \leftarrow A \cup \{e\}$