Minimum Spanning Tree

Def: **Tree** - a connected acyclic graph

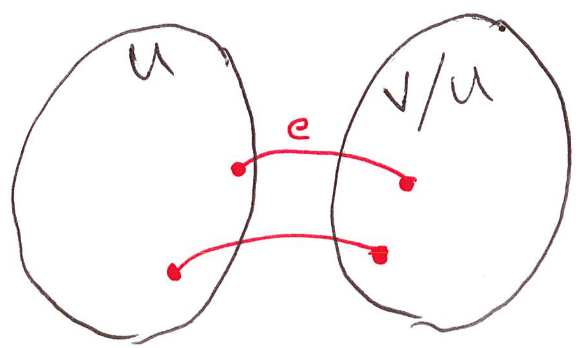
- any connected graph with $|E| = |V| - 1$

Def: **Spanning Tree** - a connected acyclic subgraph of G

Given a connected undirected weighted graph G with vertices V and edges E with a function $w : E \to \mathbb{R}$ that assigns a weight to each edge. Find a minimum weight spanning tree for G.

Greedy Choice Property

- Any non-empty proper subset U of V of G defines a cut of G into two sets U and V/U
- A cut edge is any edge with one endpoint in U and the other is in V/U
  - such an edge crosses the cut



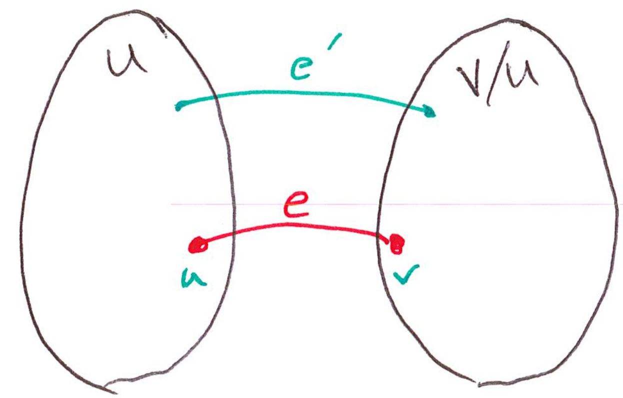
- For any cut defined by a subset U of the vertices of G, the MST for G contains the minimum weight cut edge.

<u>Proof</u>

- Let G, U, e where $e = (u, v)$ $u \in U$ and $v \in V/U$ is the minimum weight cut edge
- Let T be an arbitrary spanning tree of G that does not contain e.
- since T spans G there is a path from u to v in T not involving e

- Because $u \in U$ and $v \in V/U$ there must be at least one edge e' on this path that crosses the cut

- Remove e' from T splits T into two trees T_1 and T_2 where $u \in T_1$ and $v \in T_2$



- Adding e to T_1 and T_2 gives us a new spanning tree $T' = T - e' + e$

- since e is the lightest weight edge crossing the cut

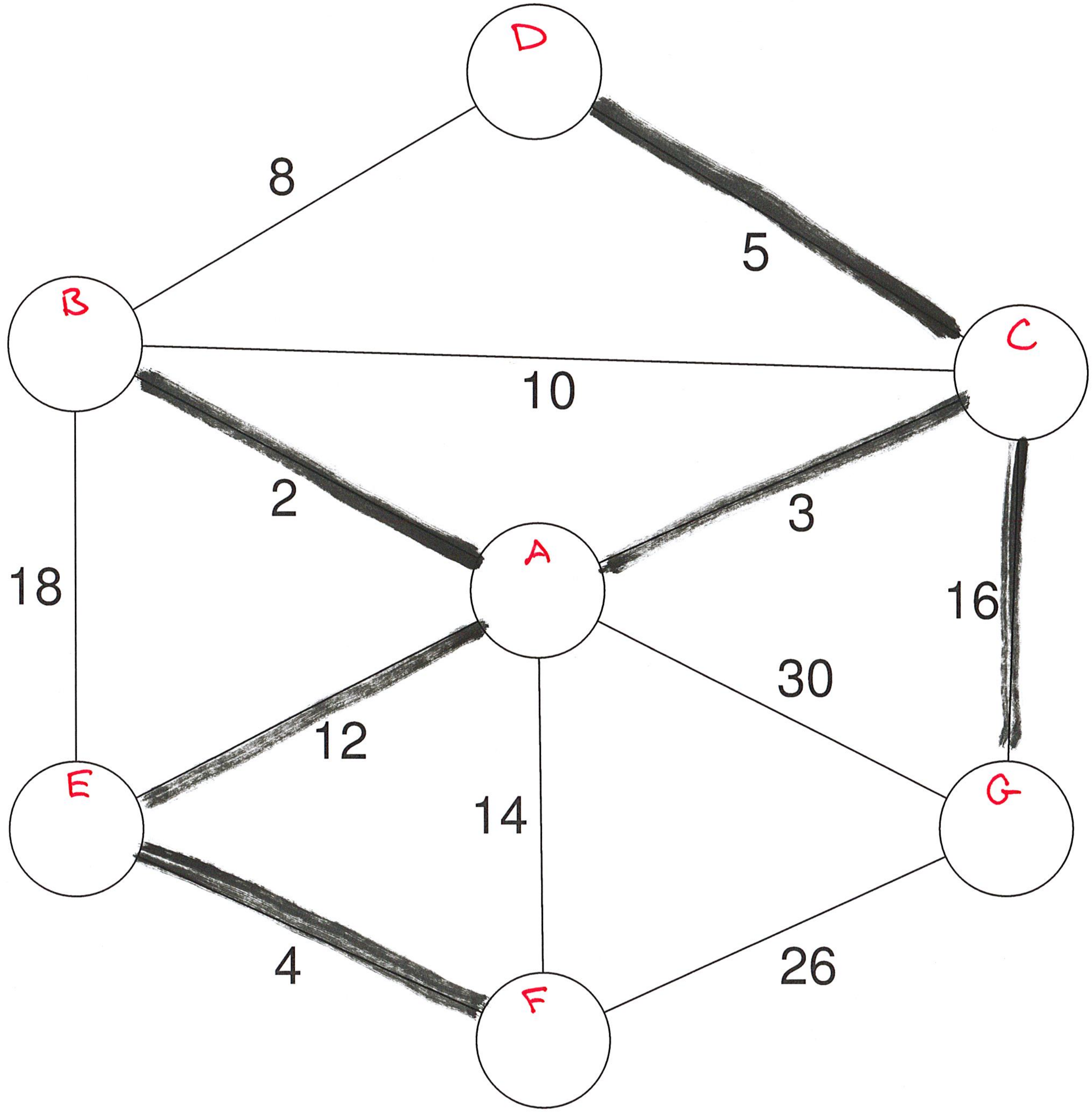
$$w(e) < w(e')$$

- so $w(T') < w(T)$

- thus T is not the MST

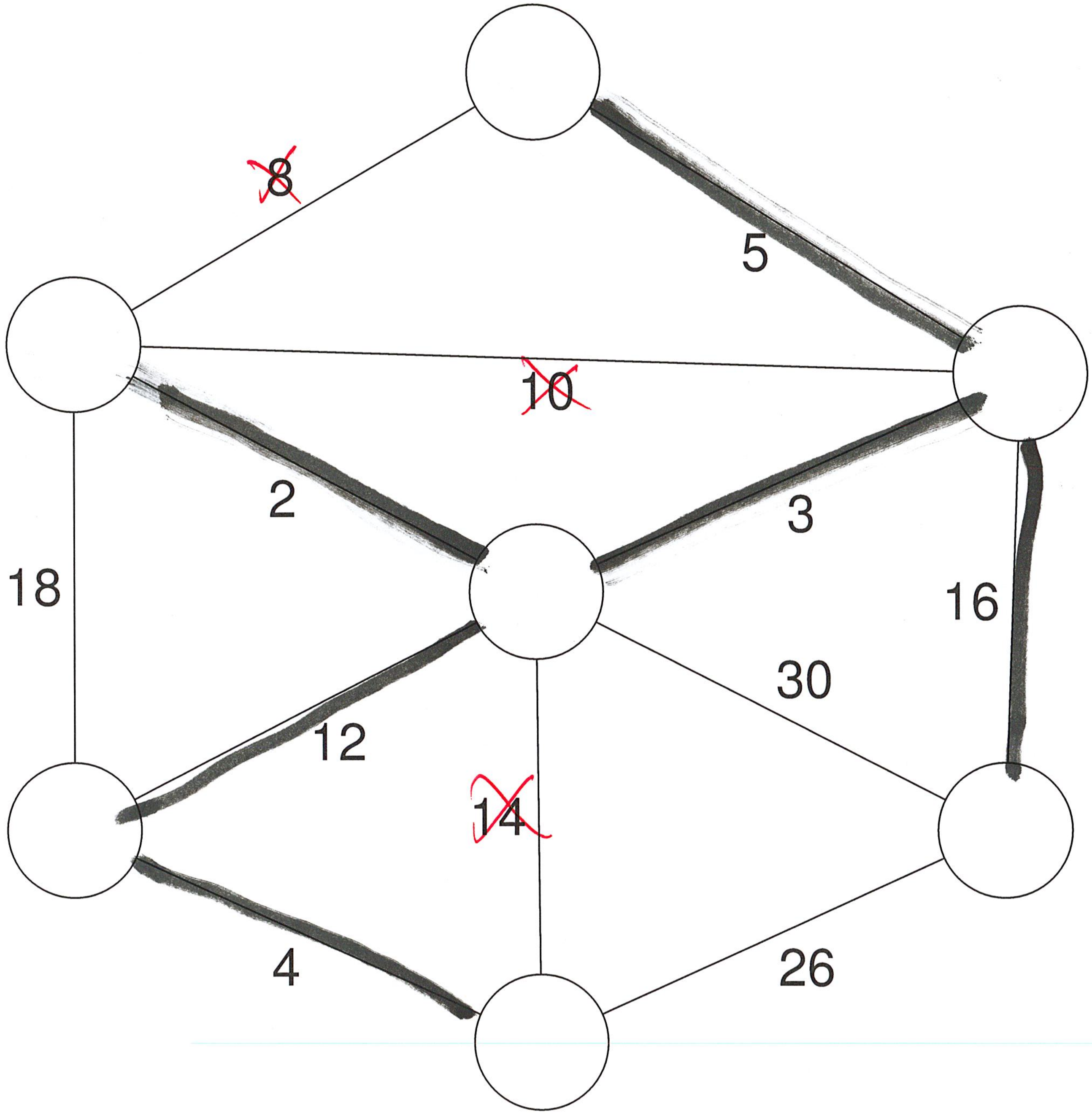
Prim's Algorithm

- our initial cut is any single vertex
- repeatedly add the lightest weight edge for the cut defined by the vertices of T until T spans the graph


D
8
5
B
C
10
2
3
A
18
16
30
12
E
G
14
4
F
26

Kuskal's Algorithm

- grow multiple trees in the spanning forest
- start by sorting the edges E by increasing weight
- for each edge add it to the forest iff its endpoints are in different trees

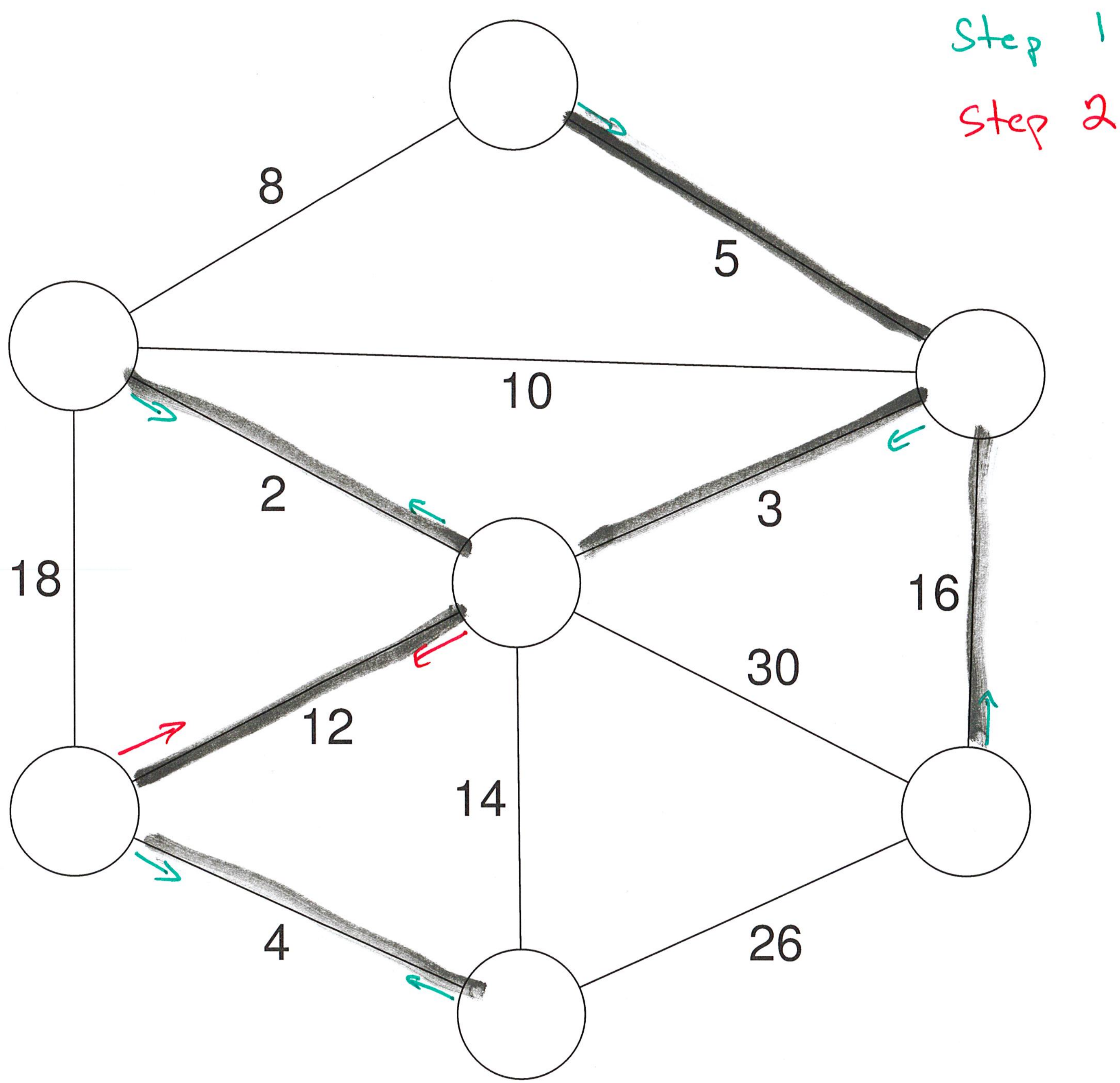

8
5
10
2
3
18
16
30
12
14
4
26

Boruvka's Algorithm

- build the intermediate spanning forest by adding the minimum weight cut edge to every component at once. Then iterate.

- At every iteration do the following
  - count the number of components C in F and label each with its component number
  - compute an array $S[1 \ldots C]$ of edges where $S[i]$ is the lightest weight edge that crosses the cut defined by vertices of component i.

- add each edge in $S[1..C]$ to F


Step 1
Step 2
8
5
10
2
3
18
16
30
12
14
4
26