# Ex. Session #1

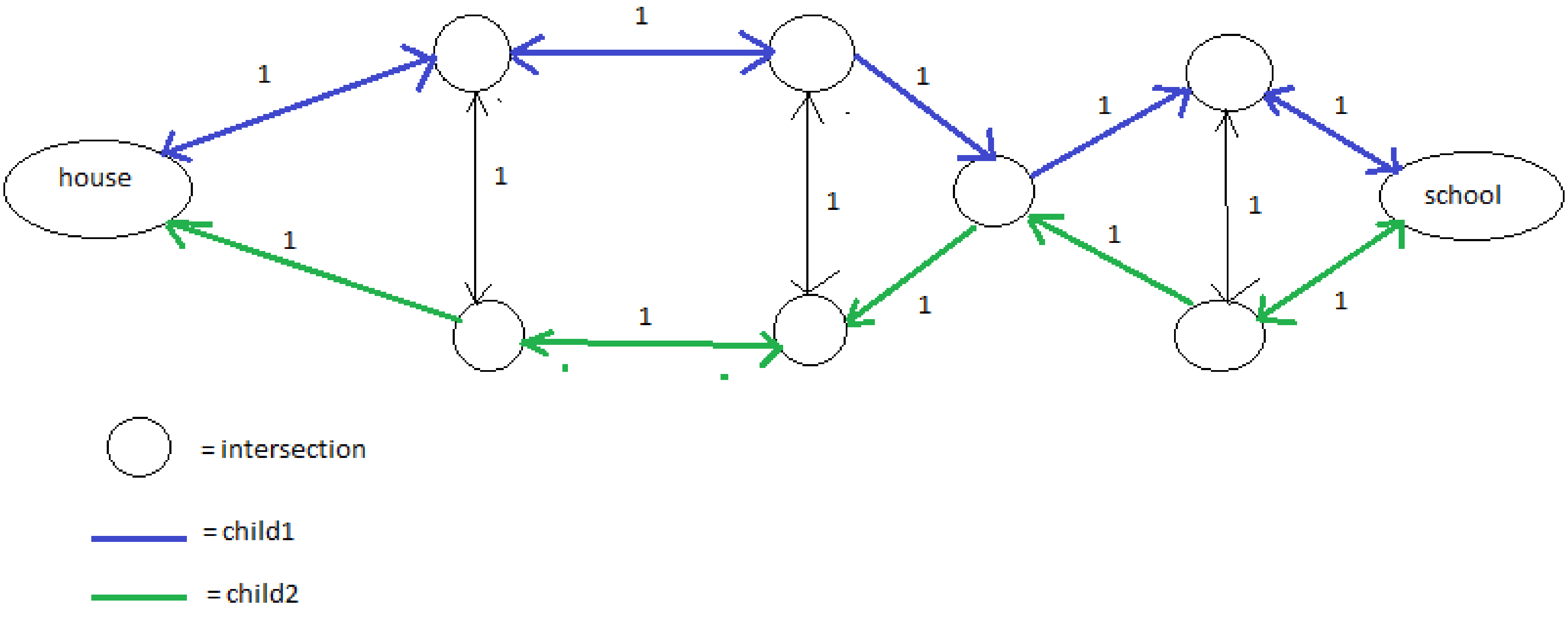
Advanced algorithms

# 26.1.6 and 21.1.7

Page 714

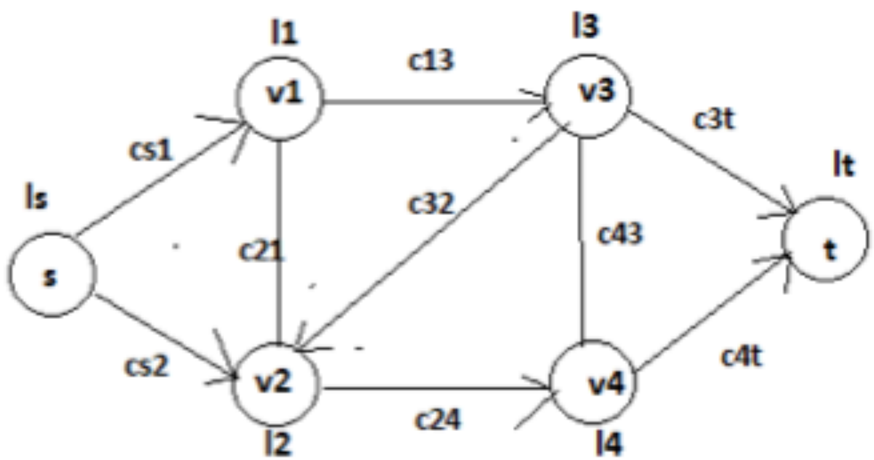
Recap: what defines a flow? What defines its value?

# 26.1-6

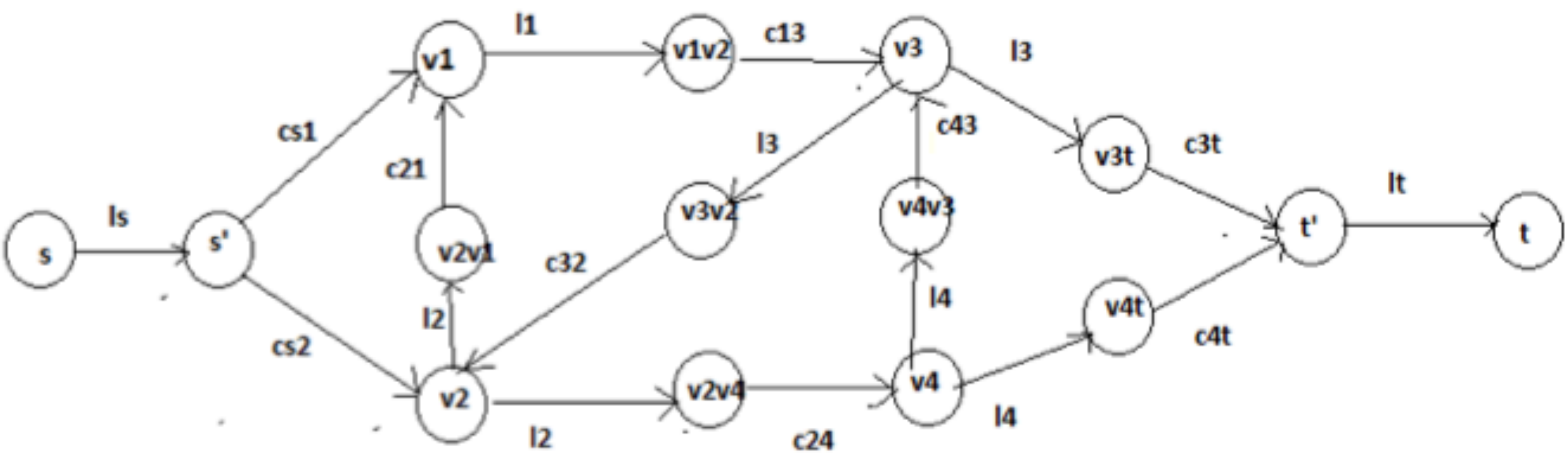



Create a vertex for each intersection/street corner and if there is a street between corner u and v create edges (u,v) and (v,u). All edges have capacity 1 and the source = house , sink = school. Find a flow of value 2 . We thus have two edge-disjoint paths from the house to the school.

# 26.1-7(2)



l1
l3
v1
c13
v3
c3t
lt
cs1
ls
c32
c21
c43
t
s
cs2
v2
v4
c4t
c24
l2
l4

l1
c13
v1
v1v2
v3
l3
cs1
l3
c43
c21
v3t
c3t
lt
ls
v3v2
v4v3
s
s'
v2v1
c32
t'
t
l4
l2
v4t
cs2
c4t
v2
v2v4
v4
l4
l2
c24

# 26.2-2

Page 710
Let’s recall

- What is a residual network?
- What is an augmenting path?
- What is a cut? What Is the capacity of the cut?

# 26.2-2

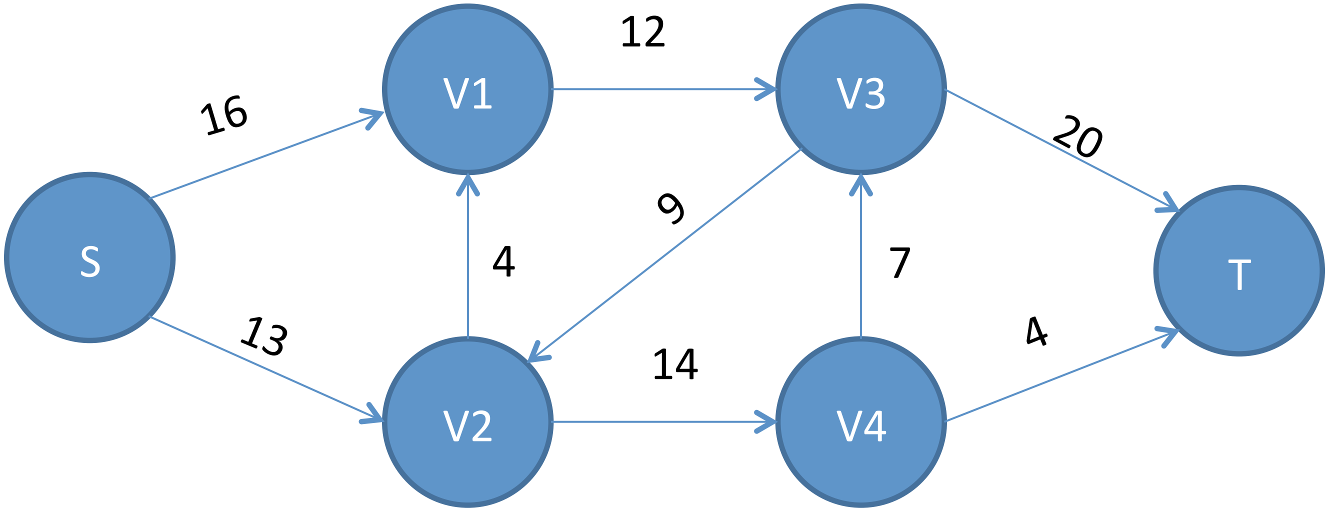
**26.2-2: Flow and capacity across cut**

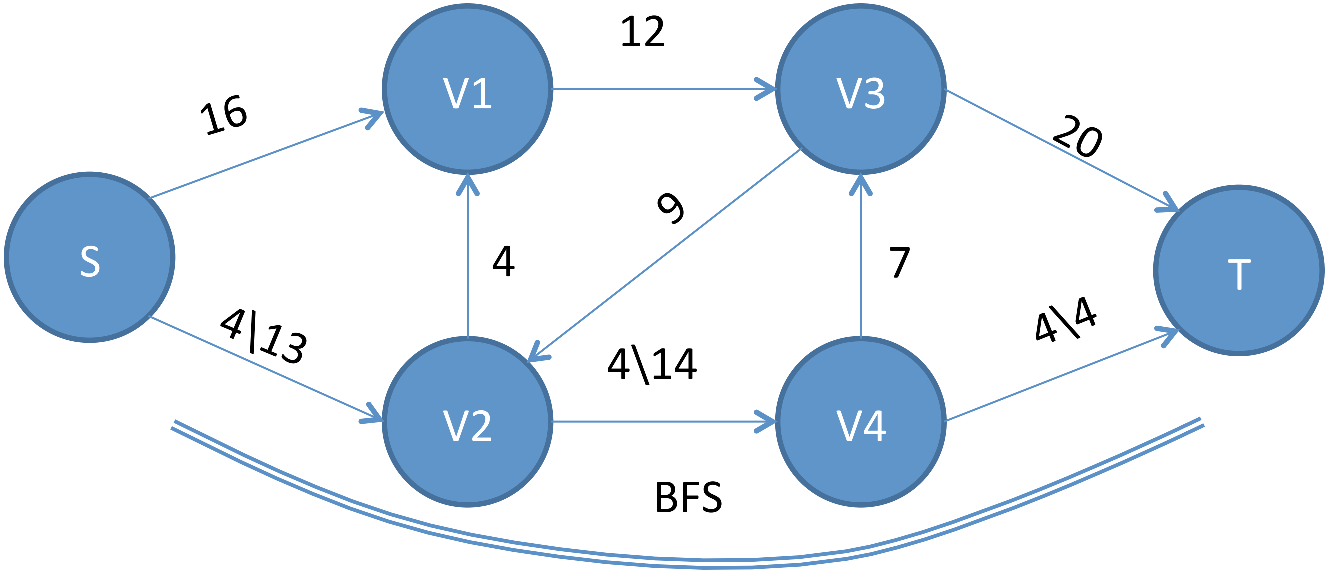
See figure on page 710, in CLRS 3rd edition.

The flow across the cut is $11 + 1 + 7 + 4 - 4 = 19$ and the capacity of the cut is $16 + 4 + 7 + 4 = 31$.

# 26.2-3 and 26.2-4

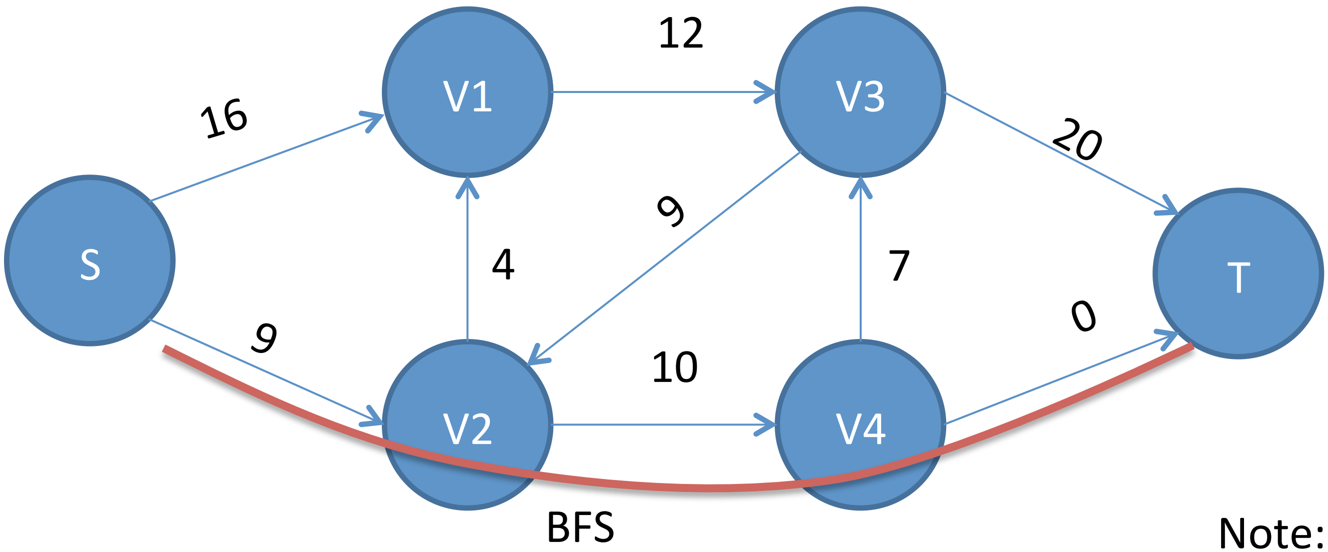
Ex.


S
V1
V2
V3
V4
T
16
13
12
4
9
14
7
20
4

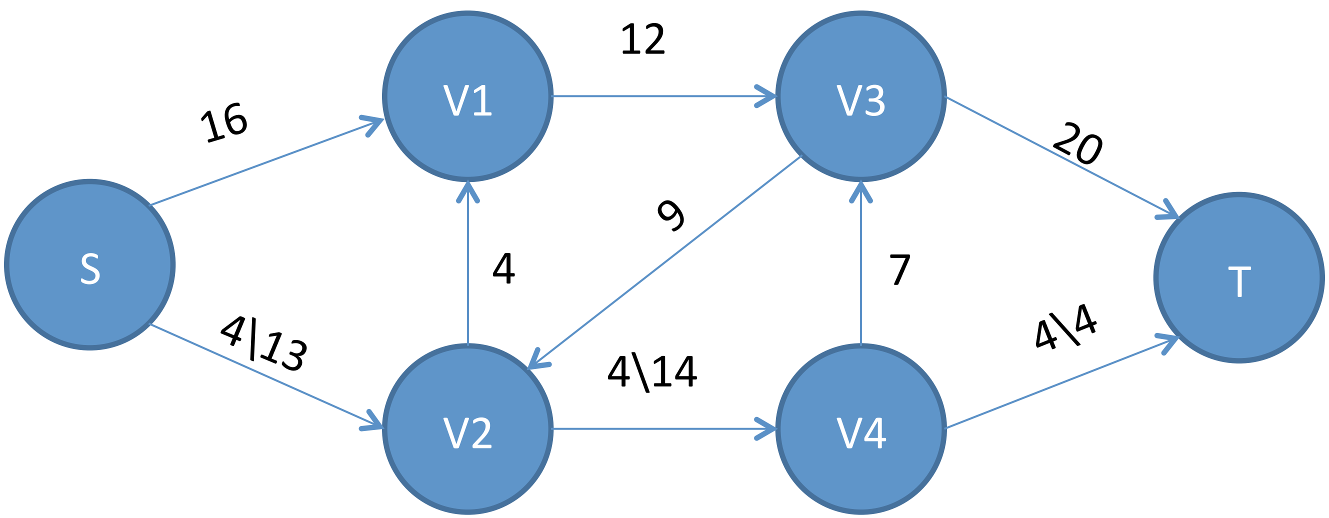


S
V1
V2
V3
V4
T
16
4|13
12
4
9
4\14
7
20
4\4
BFS

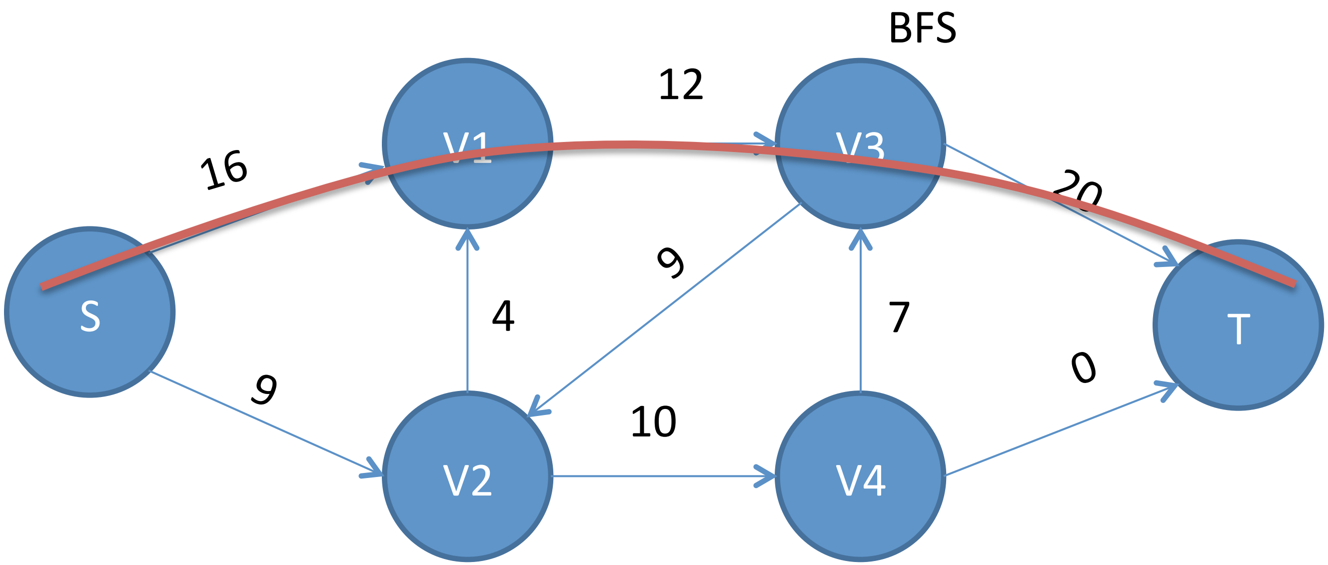
# Residual network- G f

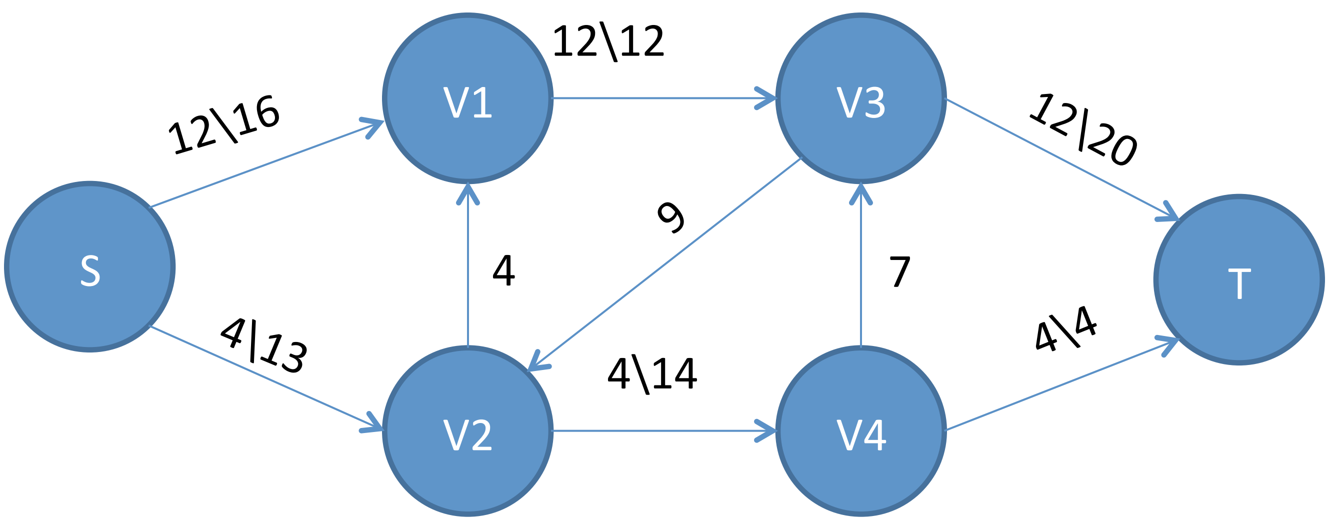



Note: No backward arrows

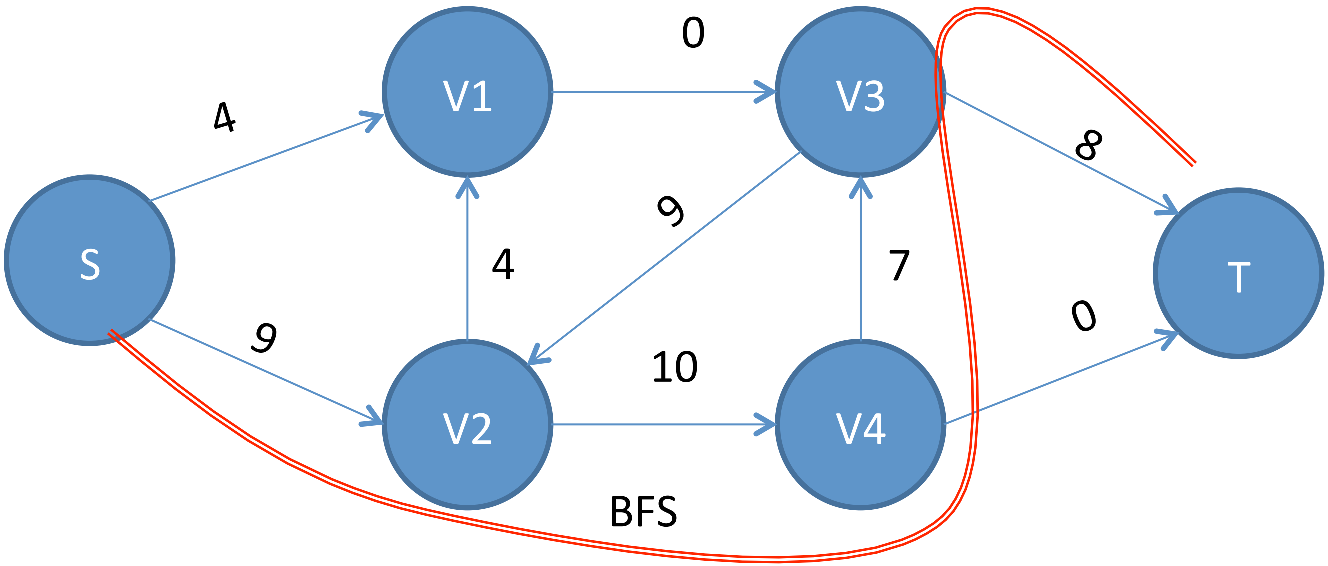


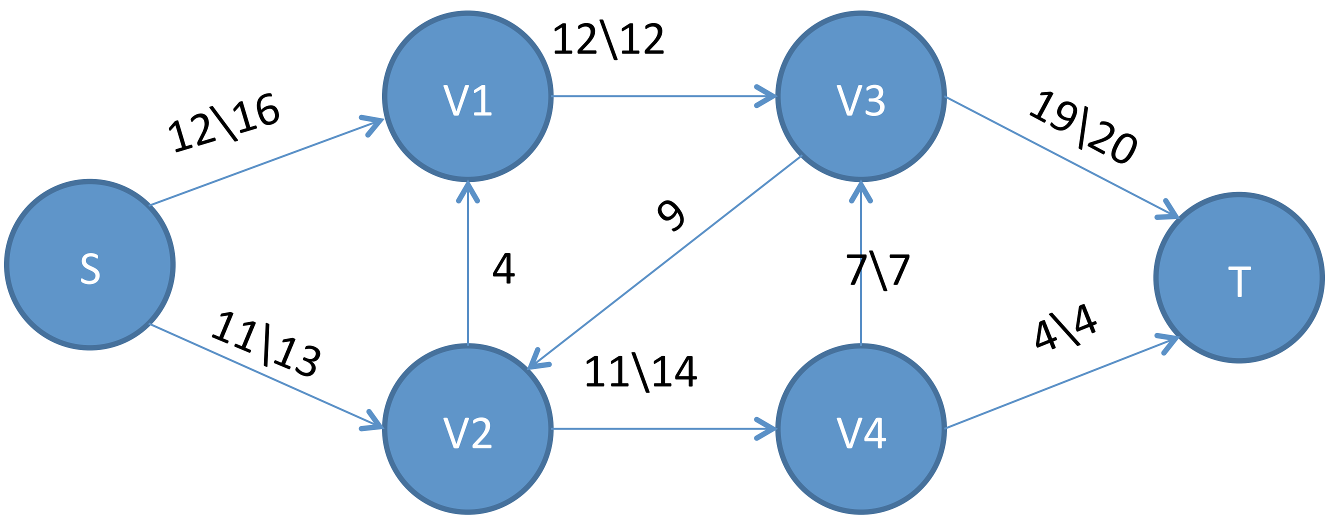
# Residual network- G f



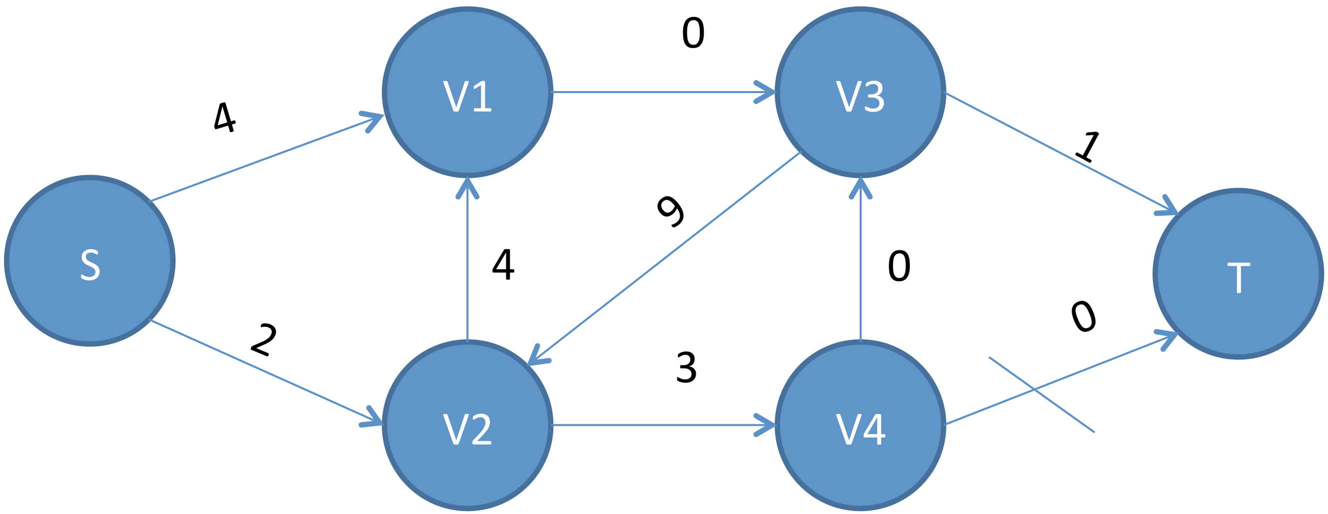



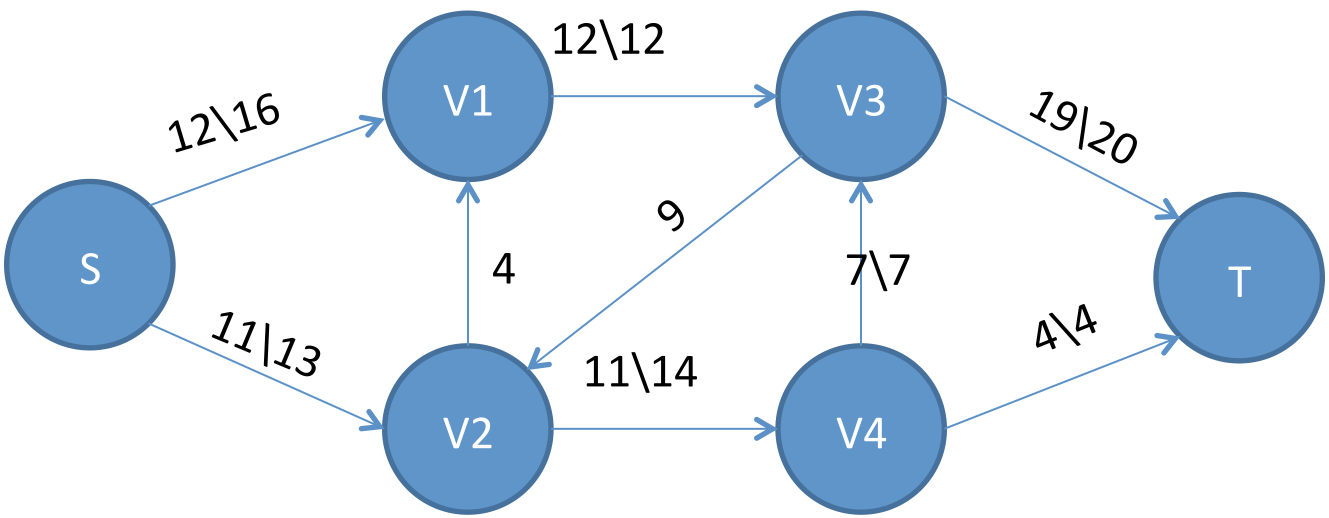
# Residual network- G $_f$



# Residual network- G $_f$

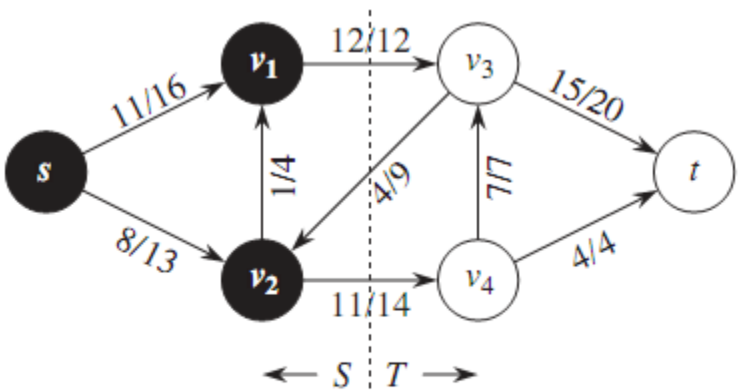


# Recall 26.2-4

The ***capacity*** of the cut $(S, T)$ is

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v) .$$

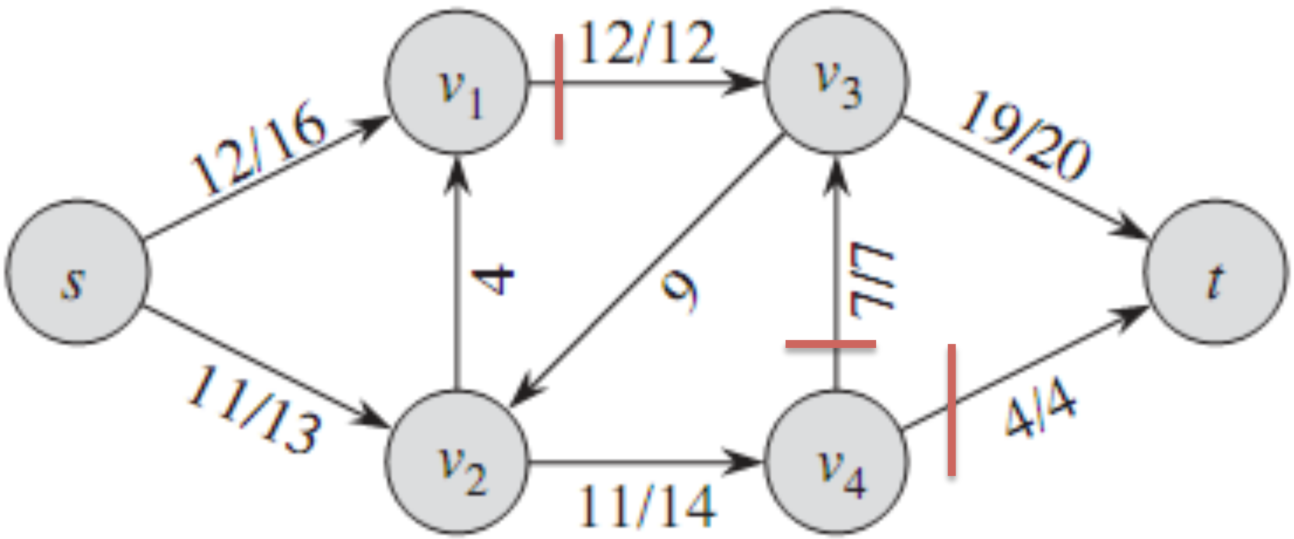



Now , we shall find the minimal

The *capacity* of the cut $(S, T)$ is

26.2-4

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v) .$$




S={s,v1,v2,v4}
T={v3,t}

# 26-4

**1.**Just execute one more iteration of Ford-Fulkerson algorithm.

Reason:

If (u,v) does not cross a minimum cut, then increasing the capacity (u, v) does not change the capacity of the minimum cut as well as maximum flow.

If (u, v) does cross a minimum cut, then it increases the capacity of the minimum cut as well as maximum flow at most 1.

In both cases we only need an augmenting path one more time.

The running time of the algorithm is O(v+e)

2.the new flow is now f'**(**u, v**) =** f**(**u, v**) -** 1.

If there is a augmenting path from u to v, then augment a unit flow along the augmenting path.

Otherwise, find augmenting paths from u to s and from t to v, decreasing a unit flow along the augmenting path.

The running time of the algorithm is O(v+e)