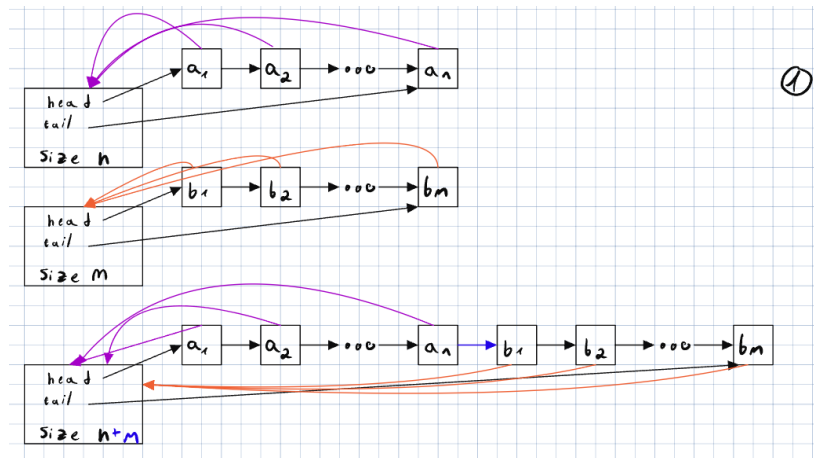


שאלה 1:



We will loop over all the items in the smaller list, and just change the pointer for the head and the tail.

```
1 def unionInner(big, small):
2
3     for item in small:
4         item.head = big.head
5         big.tail = item.tail
6     big.size += small.size
7     return big
8
9 def unionSet(s1, s2):
10     if s1.size > s2.size:
11         return unionInner(s1, s2)
12     else:
13         return unionInner(s2, s1)
```

ב:

In every step we increase the size by maximum of 2.

$$U = \{\{n_i\}: n_i \in S\}$$

אוסף של כל האברים היחידים, בעצם n פעולות .make

Join M elements with K elements where $M > K$ will take

$$O(K)$$

We prove this is section one.

If we join each group as follow, we choose the two smallest groups and join them.

We will get total of $O(\log(n))$ actions.

Where each action we be at most $O\left(\frac{n}{2}\right) = O(n)$

So we get $O(n \cdot \log(n))$

מטלה 2

6. שרשרת של ציורים יחידה ציורים צריך לקיים 3 תכונות.

$$f(u, v) = -f(v, u) \quad u, v \in V \quad \text{סמטריה ניאומית}$$

$$f(u, v) \leq c(u, v) \quad u, v \in V \quad \text{חוק הקצה}$$

$$u \in V \setminus \{s, t\} \quad \sum_{v \in V} f(u, v) = 0 \quad \text{חוק הבומר}$$

צריך לבדוק ש $f_1 + f_2$ הנה ציור חוקי.

אם קיים זוג N_{f_1} של ק"פ מטאלי בהמשך

המקוריות, אם ציור f_2 , מתקן בוויסר בהמשך

$$|f_1| + |f_2| = \text{סכום הציורים הנו} \quad \text{חוק סכום ציורים}$$

$$f_1(s, t) + f_2(s, t) = -f_1(t, s) - f_2(t, s) = -[f_1(t, s) + f_2(t, s)]$$

$$f'(s, t) = -f'(t, s) \rightarrow \underline{\underline{\text{חוק 1}}}$$

$f_1(s, t) + f_2(s, t) \leq c(s, t) \Rightarrow$ לאבד עמידה צריך שה c
 תהיה פונקציה של s בלבד.

$$\sum_{v \in V} [f_1(u, v) + f_2(u, v)] = \underbrace{\sum f_1(u, v)}_{\text{אנשים ב- } u \text{ שרוצים}} + \underbrace{\sum f_2(u, v)}_{\text{אנשים ב- } u \text{ שרוצים חוקית}} = 0$$

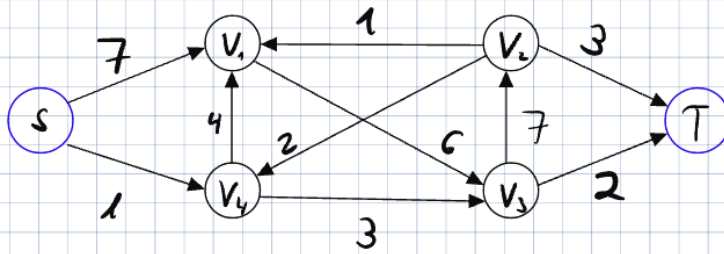
2) f_1 | f_2 ה'לם כלילתה ברשת N

בלתי-אפשרי N_{f_1} שיהיה $|f_1| - |f_2|$

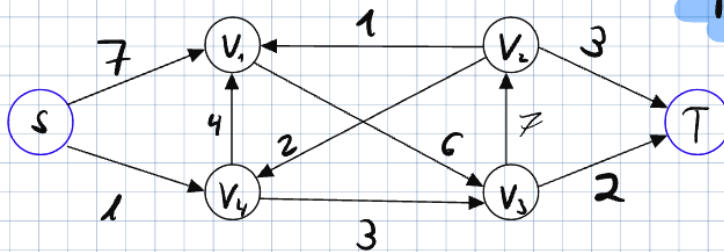
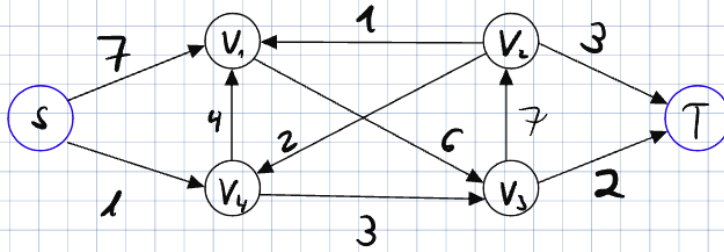
אם נרצה להכניסם שוב נצטרך פונקציה c

ברשת G אנחנו מבינים בדיוק מה
 נקבעים שכל אחד מהם יהיה

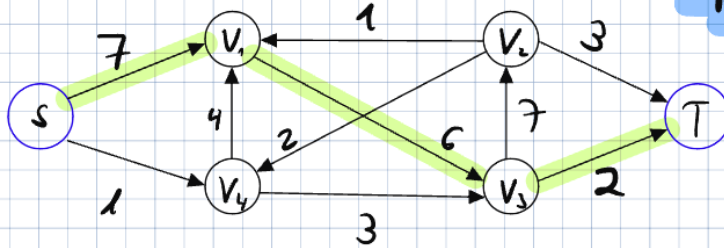
ב- $|f_2|$ ברשת N_{f_1} , כל אחד מהם
 הפונקציה f_2 היא פונקציה של f_1 .



10 3



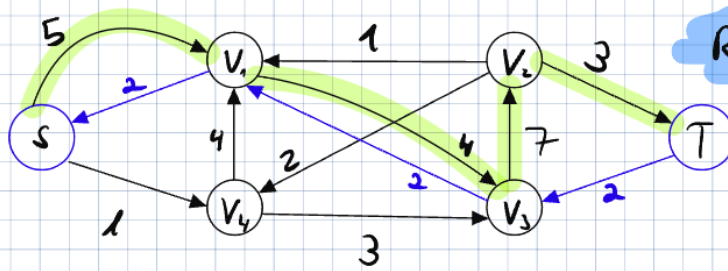
Residual network



Residual network

find a path using
DFS or BFS

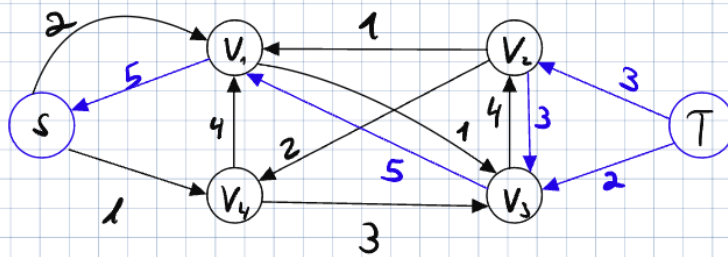
min flow - 2



Residual network

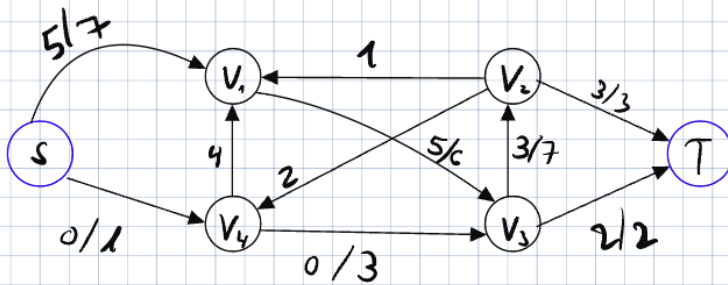
find a path using
DFS or BFS

Min flow - 3



Done There

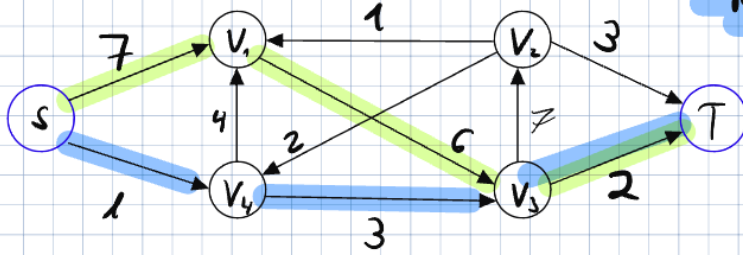
is no residual
path from S → T



max flow 5

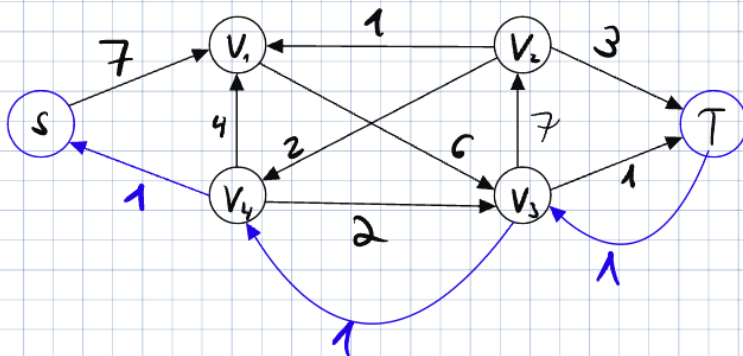
(23)

Residual network

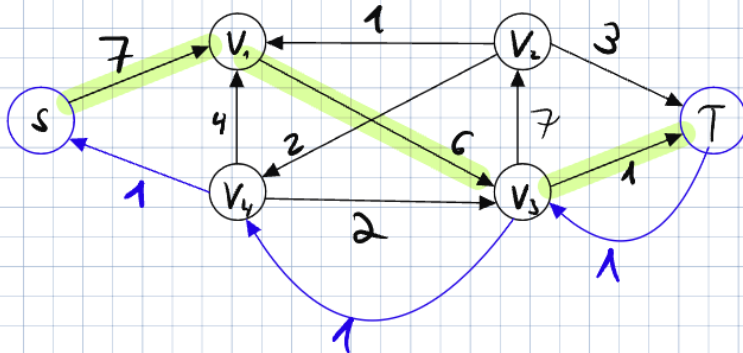


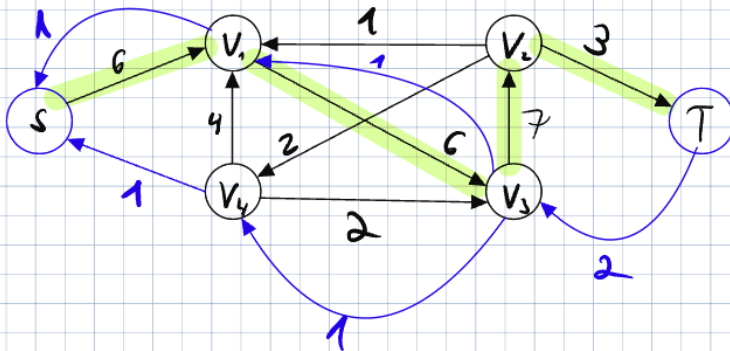
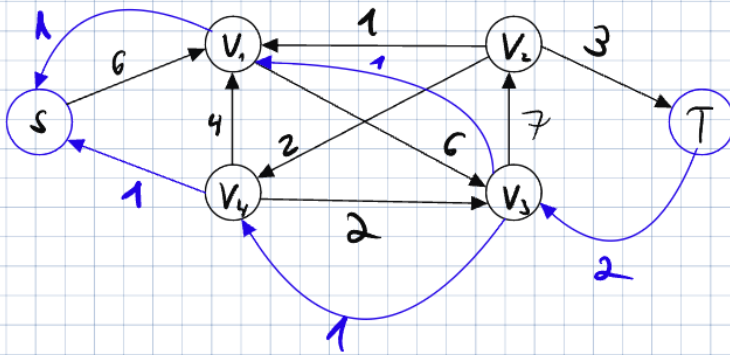
find the shortest path

We have two shortest path
We will choose min flow

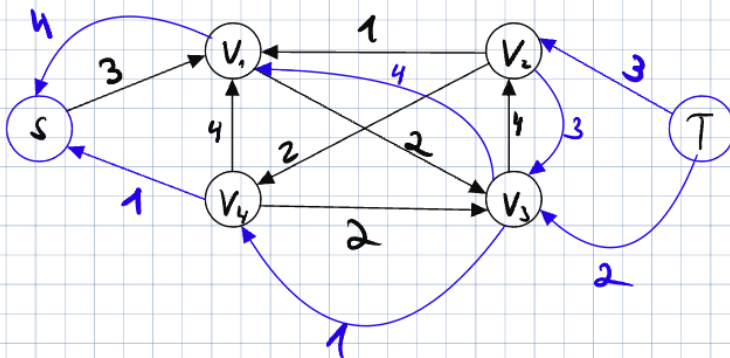


We choose as the shortest path
pass flow $\min(1)$

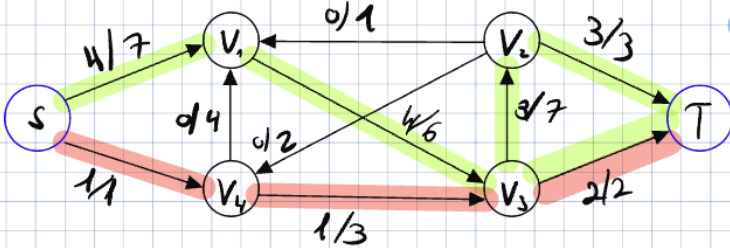




choose the current
shortest path
pass min flow (3)



No more residual path
so we are done



Done Flow 5

$$\min \text{ cut} = \max \text{ flow}$$

1037e

② ③

אמין / שפוט / סכום 5, 'ס' Ford/ER, חסר

המחיר / 5.

אפשר לראות את המרחק $\{v_2, v_3, T\}$

- 1) Create the residual graph $O(V+E)$
- 2) Find a path from the source to the target in the residual graph. If you find return False. $O(V + E)$
- 3) Else return true. $O(1)$

$$O(V + E) + O(V + E) + O(1) \rightarrow O(V + E)$$

Step 1:

Creating the residual graph, we will need to copy the graph and creating the residual while copy.

Just to note if we don't copy the graph but modify the existing one this will take $O(E)$ time.

Step 2:

Finding if there is a path using BFS or DFS will take $O(V+E)$

Total running time is $O(V+E)$