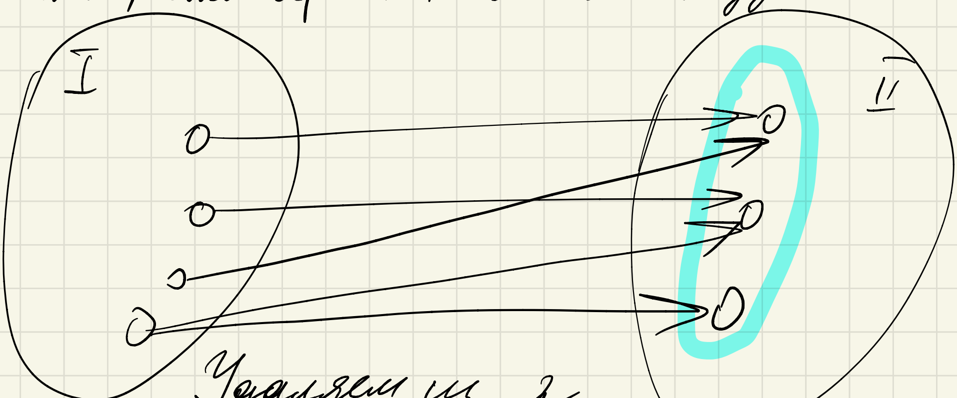
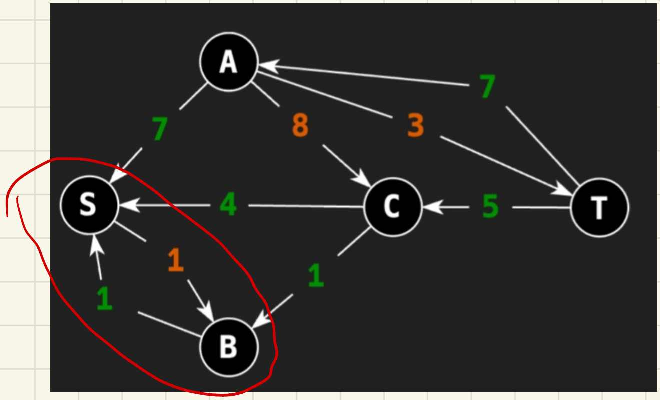
Задачу поиска минимального числа вершин, которые необходимо удалить, чтобы из S стало невозможно попасть в T сведём к задаче поиска минимального разреза, которую сведём к поиску макс. потока.

Итак, если мы найдём минимальный разрез, то, посчитав число уникальных вершин из 1ой и 2ой компоненты связности и взяв минимум среди них, мы найдём мин. число вершин, которые необходимо удалить. Заметим, что вершины s и t мы удалять не можем

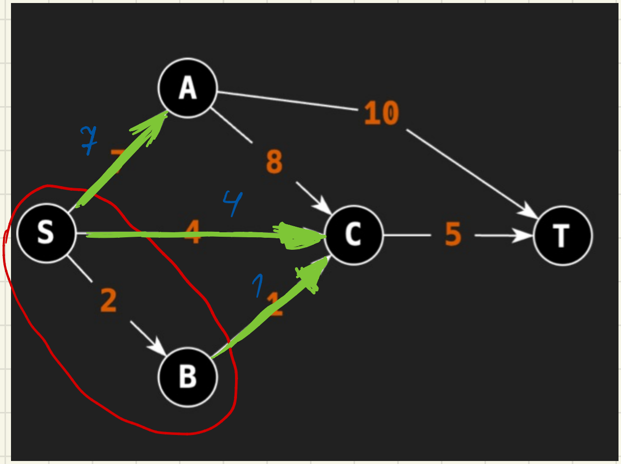


Удаляем из 2 компоненты, т.к. там меньше (отв: 3)

Докажем, что задача поиска макс. потока двойственна задаче поиска мин. разреза. Пусть в результате работы алгоритма Форда-Фалкерсона получилась такая остаточная сеть:



Выделим вершины, достижимые из истока в ост. сети. Теперь возьмем исходный граф и отметим все рёбра, выходящие из выделенной компоненты. Эти рёбра и будут явл. минимальным разрезом графа



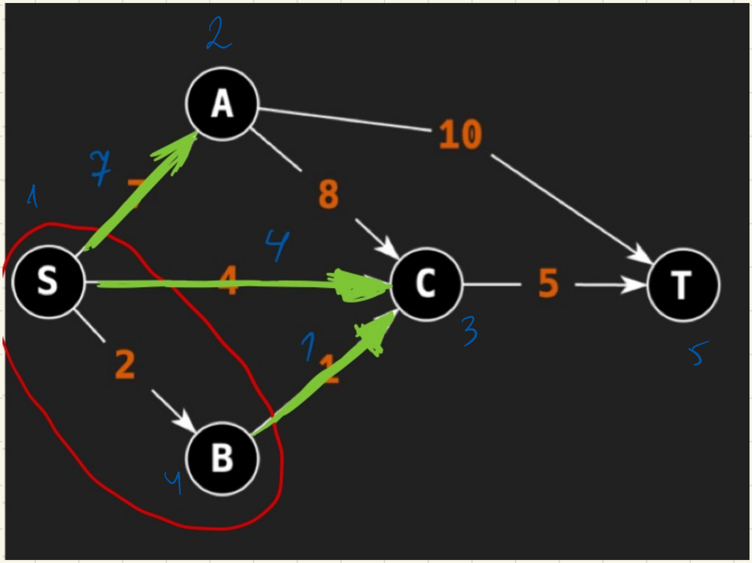
Сумма мин. разреза: $7+4+1=12$ и равна сумме макс. потока

Докажем, что такой разрез минимален. В целом, для любого разреза C и потока F выполняется: $|F| \le |C|$ — поток никак не может превысить пропускную способность разреза. Значит, для максимального потока выполняется такое же неравенство: $|F_{max}| \le |C| \Rightarrow$

$\Rightarrow$ если мы среди всех разрезов выберем минимальный, то неравенство превратится в равенство: $|F_{max}| = |C_{min}|$

Тогда, сформировав множество вершин, достижимых в ост. сети из истока, мы гарантируем, что в ост. сети в это множество только входят рёбра, а значит, величина потока через них равна их пропускной способности. Также мы гарантируем, что найденный список рёбер будет разрезом, т.к. в исходном графе эти рёбра будут связывать две компоненты связности, значит, удалив их, граф перестанет быть связным. Таким образом, раз величина потока через каждое ребро равна его пропускной способности, то поток через весь разрез явл. максимальным в графе, а значит, разрез является минимальным.

чтд



Таким образом, в этом графе надо удалить вершины A и C
2 3

Результат работы алгоритма:

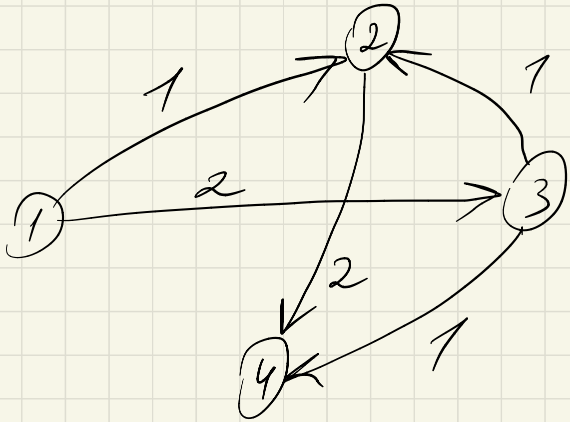
Исходный граф на картинке

Ответ на него:

```
Residual network at the end:
0 0 0 1 0
7 0 8 0 3
4 0 0 1 0
1 0 0 0 0
0 7 5 0 0
Vertices to delete: 3 2
```

Проверим также на тестах из задачи P6:

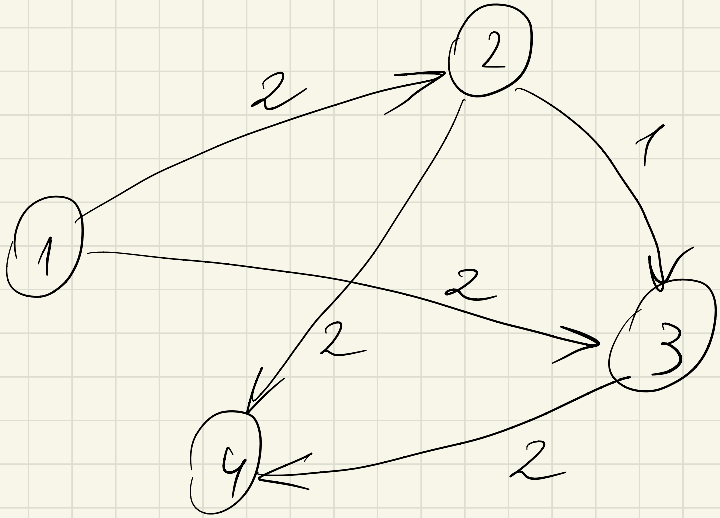
Пример 1:



Ответ:

```
Residual network at the end:
0 0 0 0
1 0 1 0
2 0 0 0
0 2 1 0
Vertices to delete: 3 2
```

Пример 2:



Ответ:

```
Residual network at the end:
0 0 0 0
2 0 1 0
2 0 0 0
0 2 2 0
Vertices to delete: 3 2
```