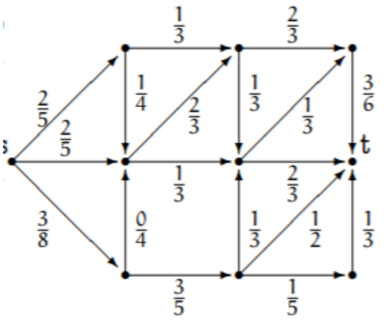
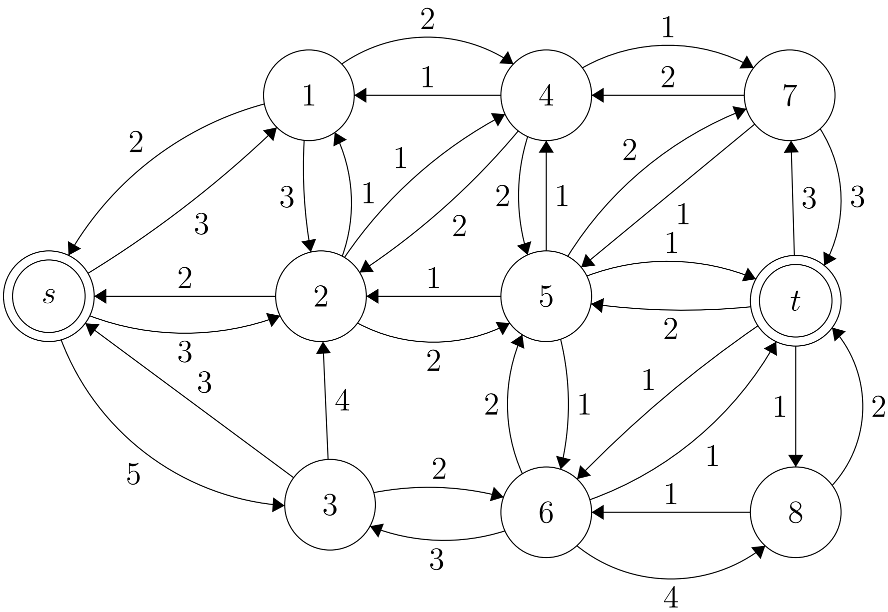
Филиппова Анастасия

Домашнее задание 9

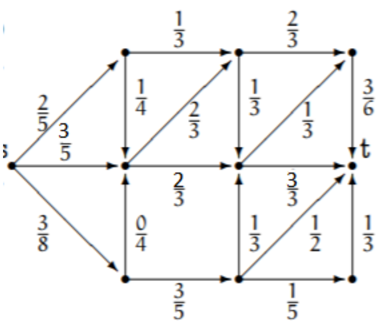
1



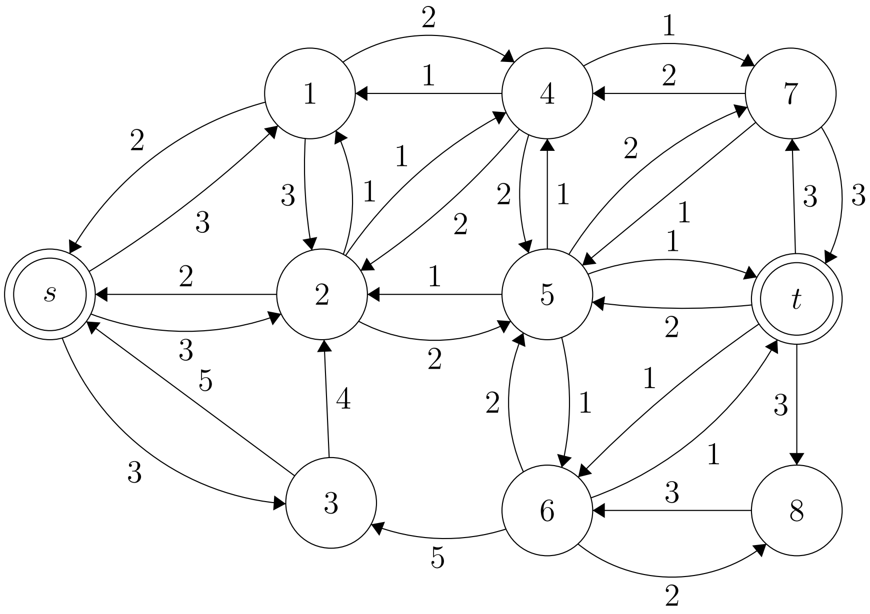
(i) $f = 7$ (сумма потоков из источника).
(ii)



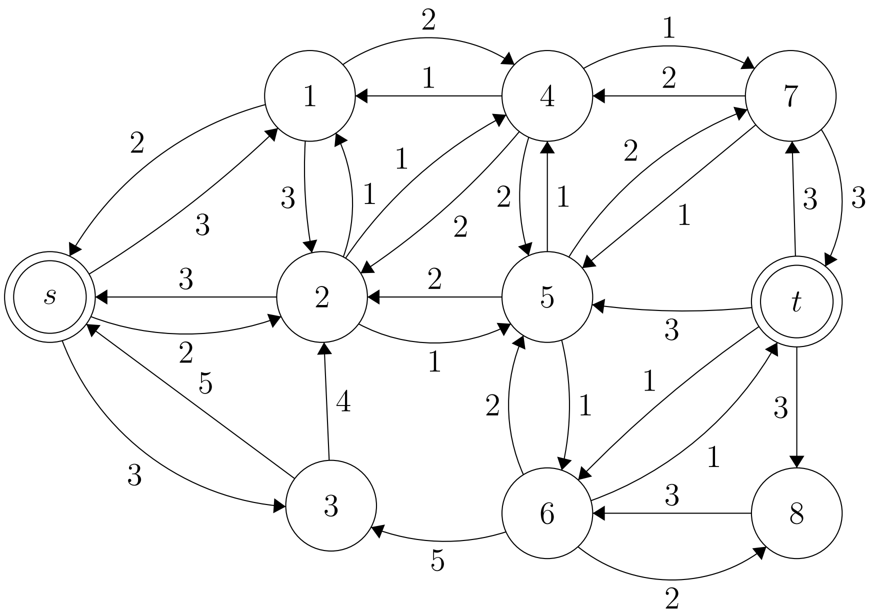
(iii) Очевидно, что поток не максимальный, так как можно увеличить поток через одно из ребер и сделать поток равным 8:



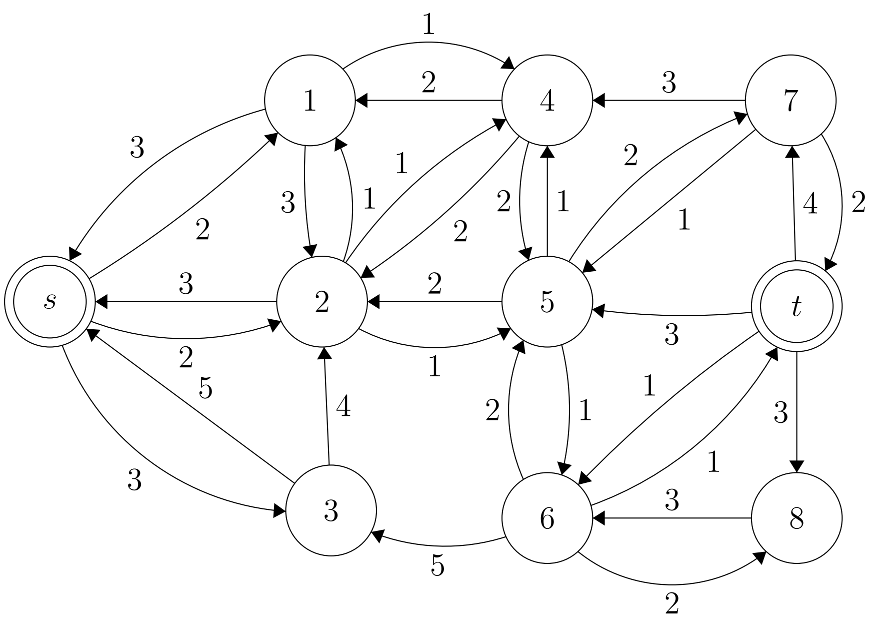
(iv) $1) s - 3 - 6 - 8, c_{min} = c_{36} = 2.$



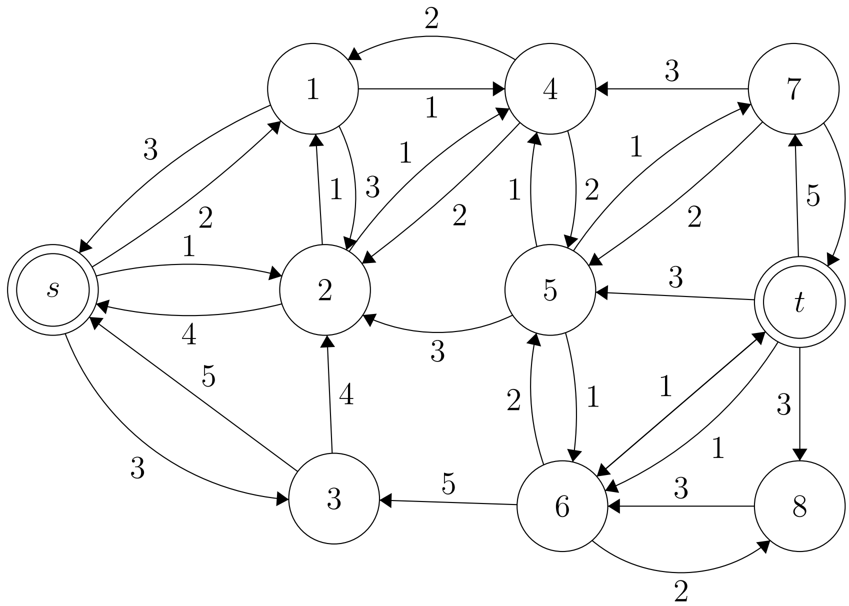
$2) s-2-5-t, c_{min}=c_{5t}=1.$



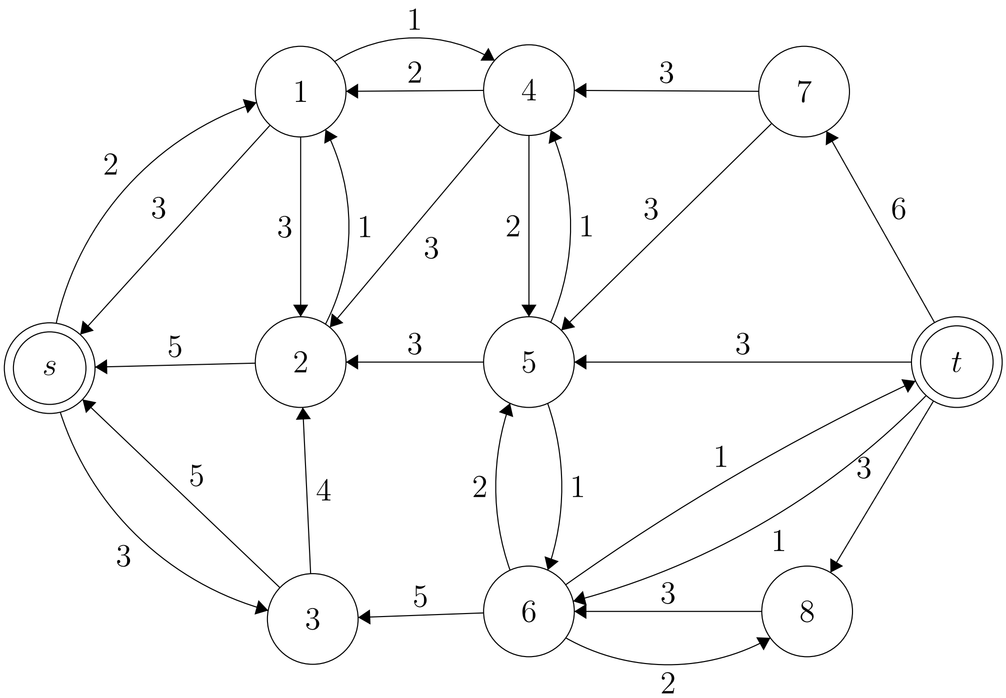
$3) s-1-4-7, c_{min}=c_{47}=1.$



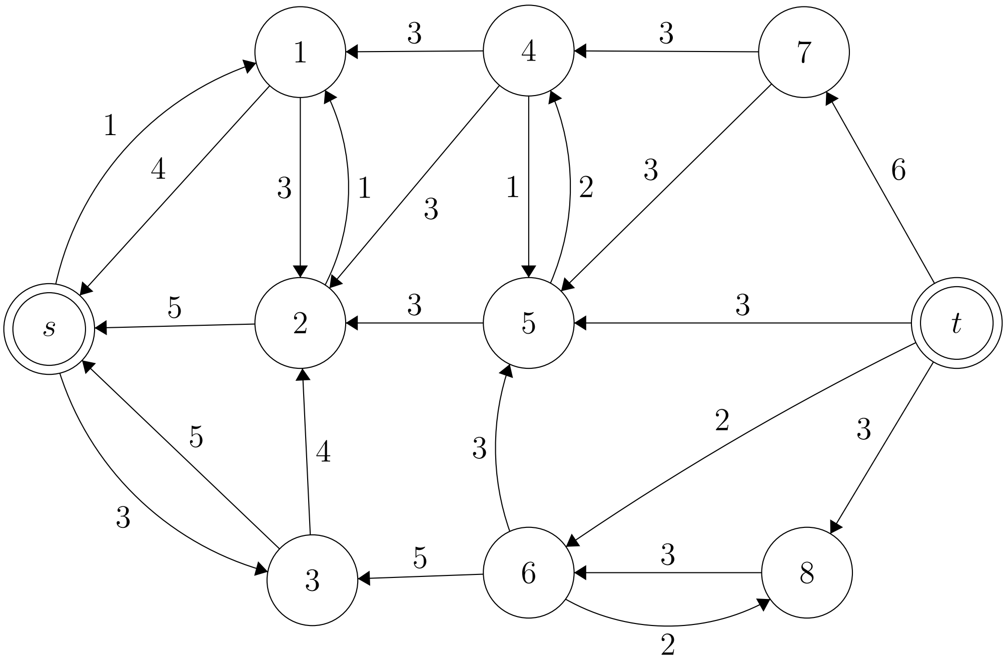
$4) s-2-5-7-t, c_{min}=c_{25}=1.$



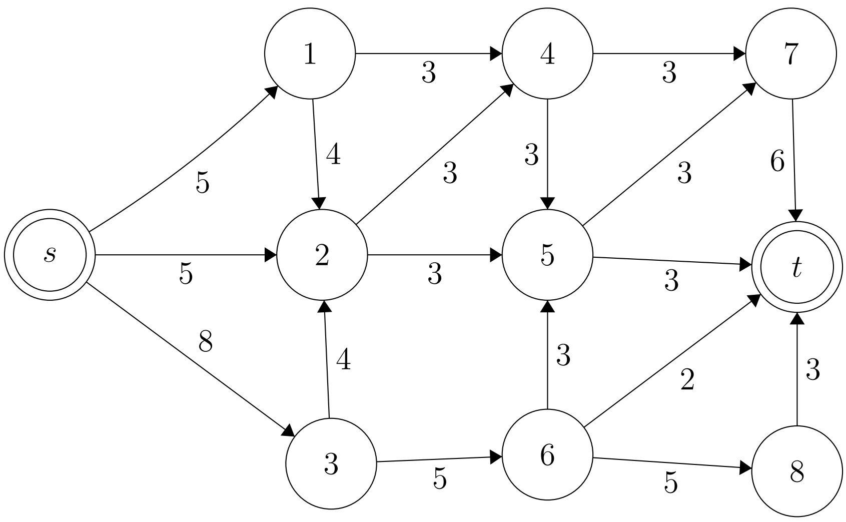
$5) s-2-4-5-7-t,$



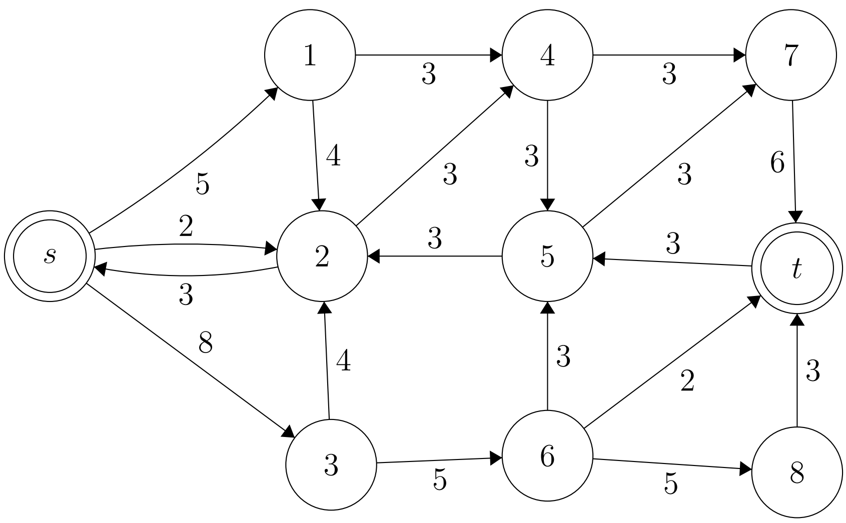
$6) s-1-4-5-6-t, c_{min}=1,$



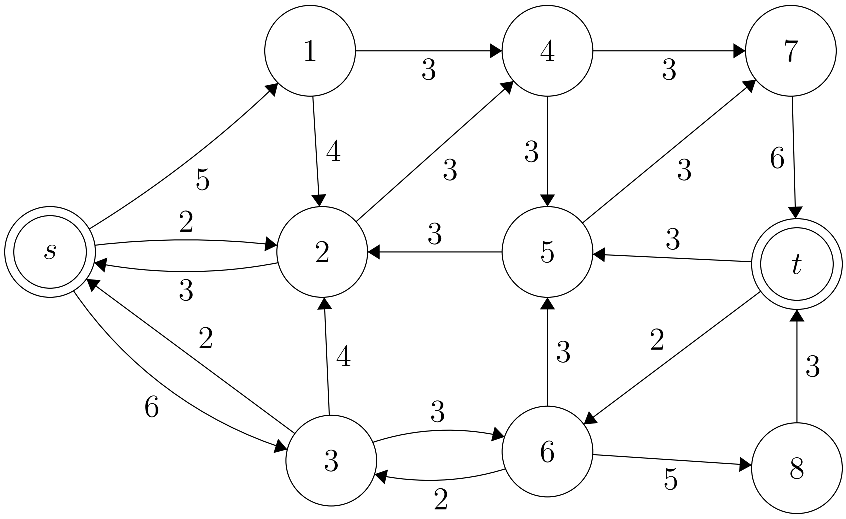
Минимальный разрез: $(4,1),(4,2),(5,2),(6,3) \iff f_{max} = 3+3+3+5 = 14$.
(v) Обнуляем все потоки. Остаточная сеть изначально совпадает с исходной сетью.



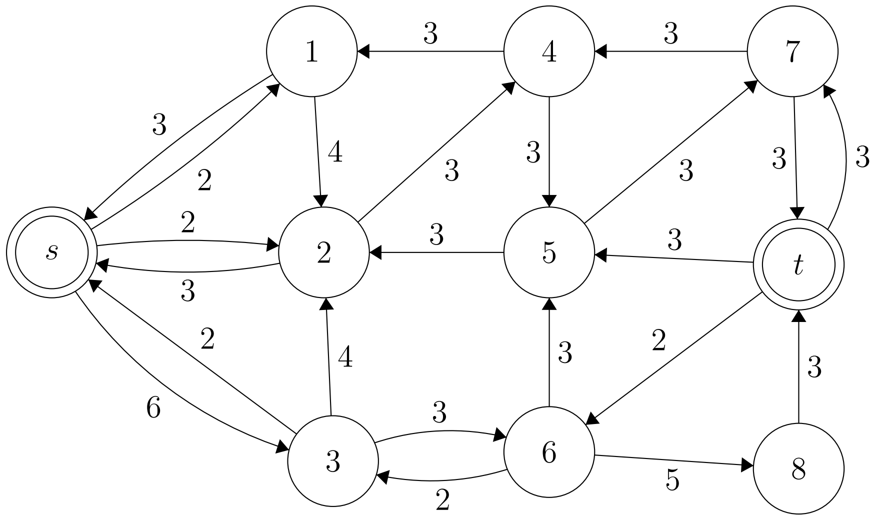
Теперь находим кратчайшие пути с помощью BFS :
$1)s-2-5-t, c_{min} = 3,$



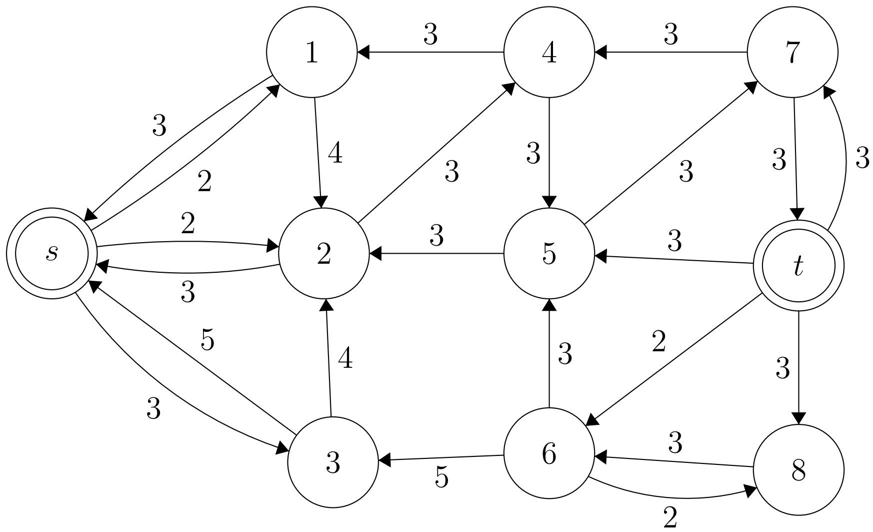
$2) s-3-6-t, c_{min}=2,$



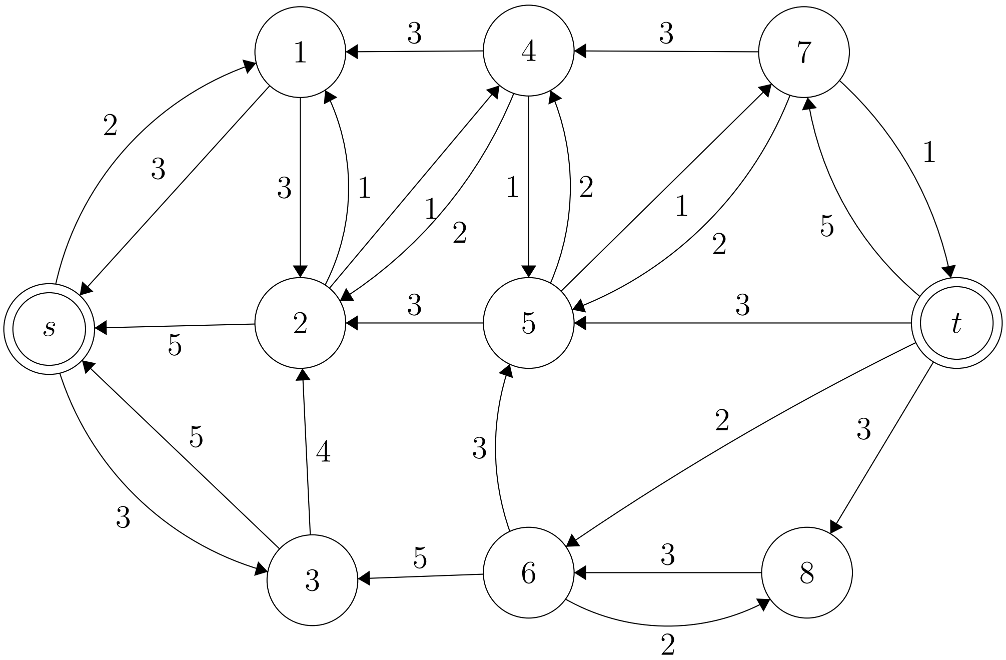
$3) s-1-4-7-t, c_{min}=3,$



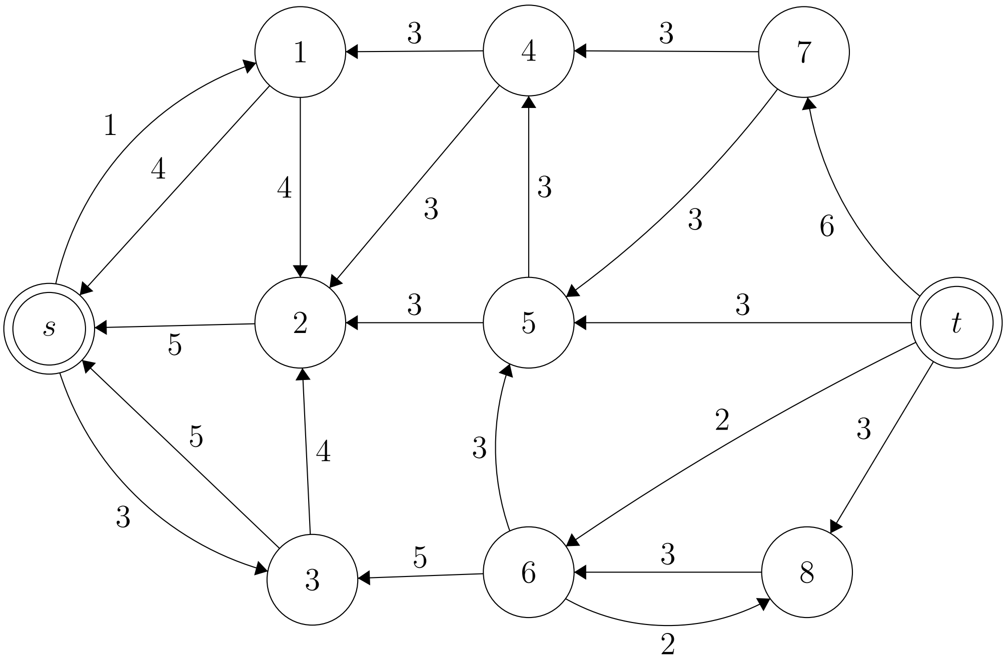
$4) s-3-6-8-t, c_{min}=3,$



$5) s-2-4-5-7-t, c_{min}=2,$



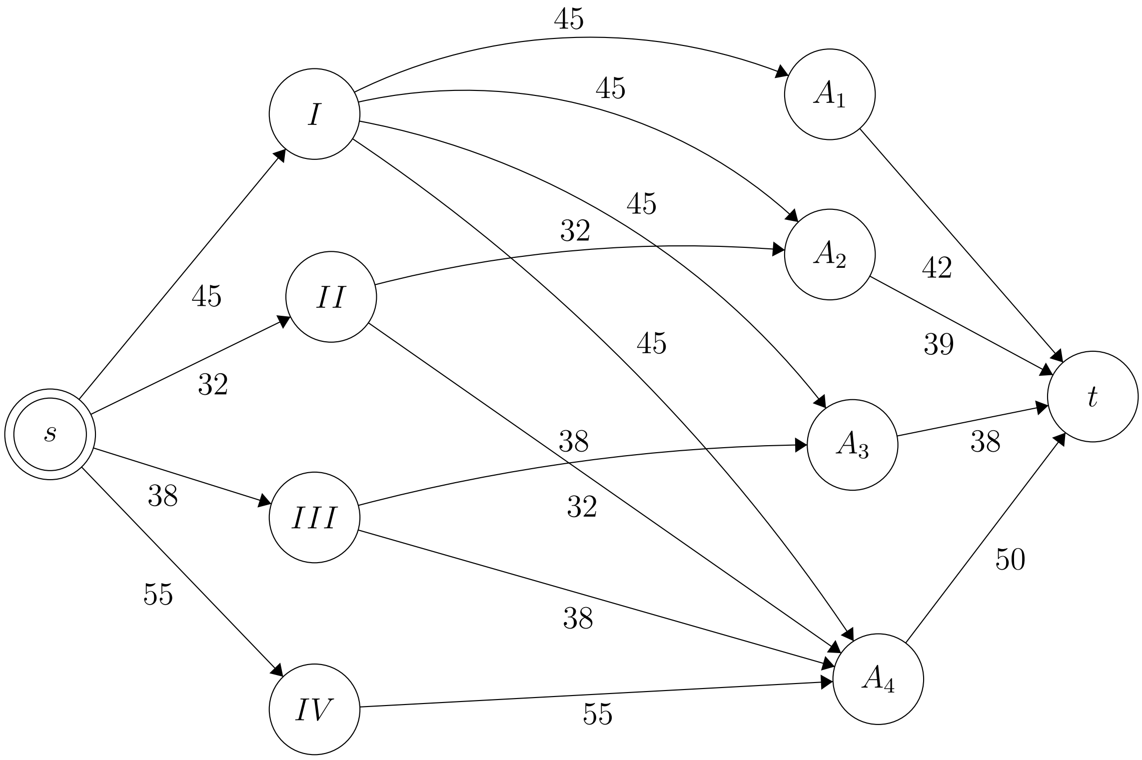
$6) s - 1 - 2 - 4 - 5 - 7 - t, c_{min} = 1,$



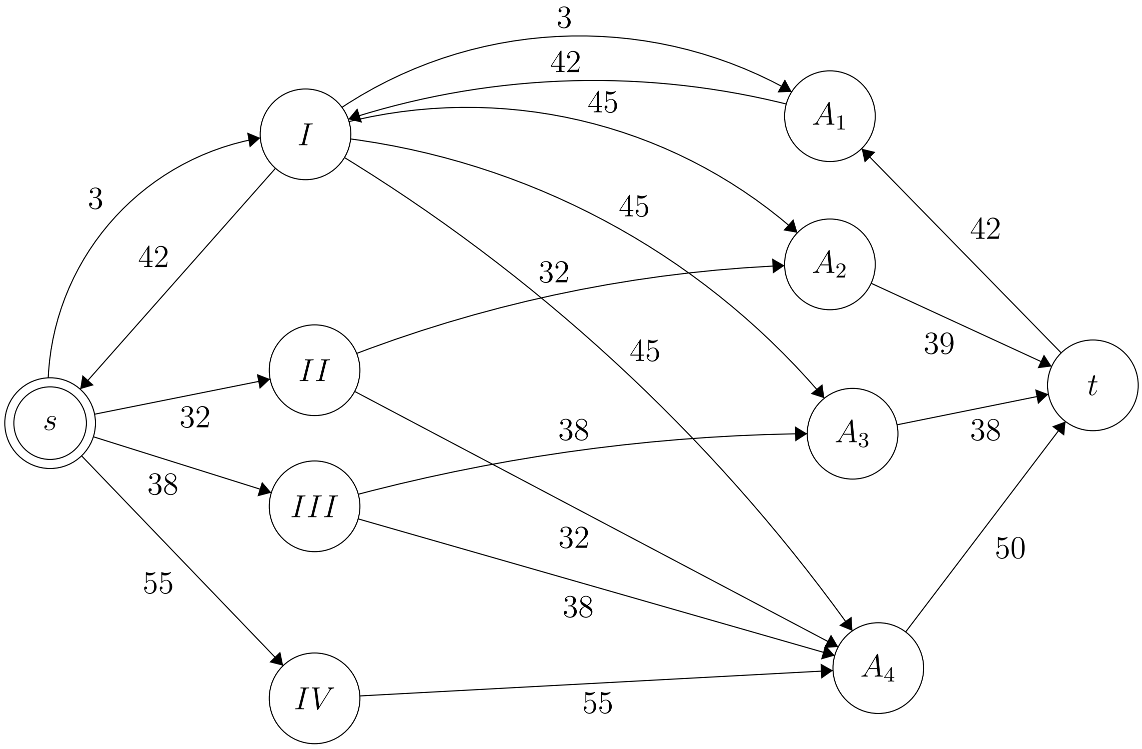
Мы получили тот же минимальный разрез, $f_{max} = 14$.
Ответ: 14.

2

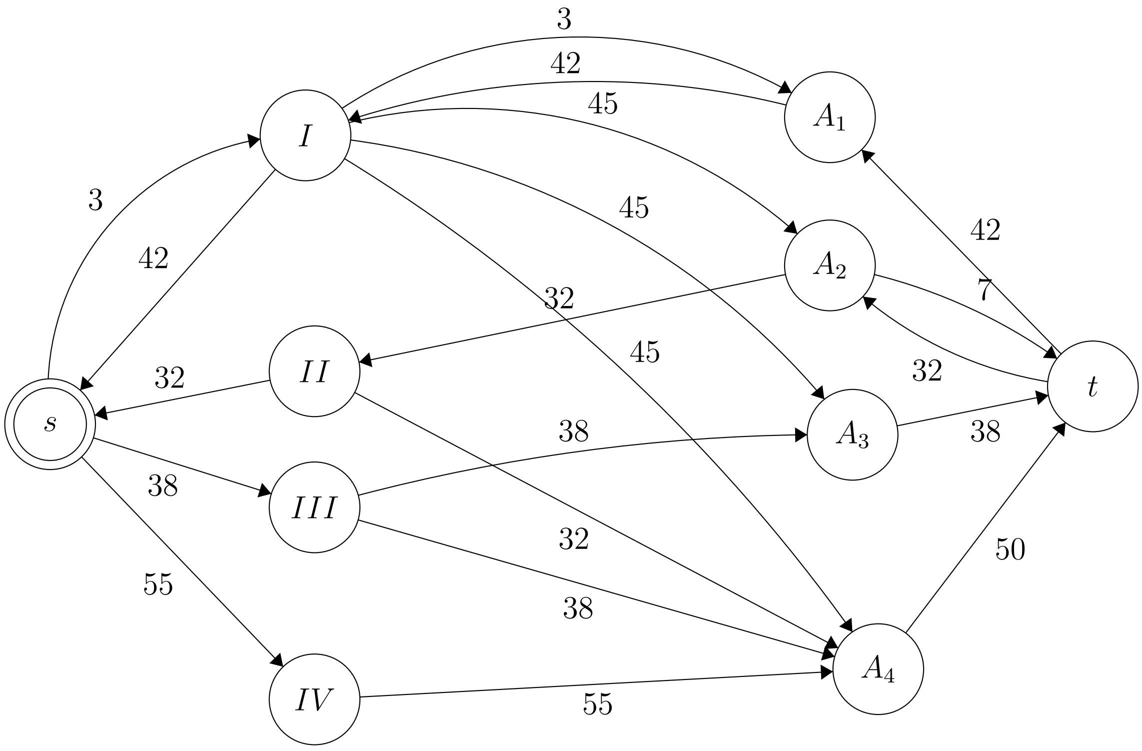
(i) Решим данную задаче с помощью нахождения максимального потока. Представим данные ввиде графа, изображенного ниже. Для каждой группы создадим по 2 вершины так, как показано на рисунке и соединим их ребрами в соответсвии с условием. Из вершины s направим ребра во все вершины с пропускными способностями равными наличию доз для каждо группы крови соответсвтенно. Из вершин I направим ребра с пропускной способностью 45 во все $A_i, i = 1, .., 4$, так как пациентам с остальными группами крови можно переливать кровь первой группы. Аналогично поступаем с остальными вершинами II, III, IV. В вершину t напрявляем ребра из всех $A_i, i = 1, ..., 4$ с пропускной способностью равной запросу для каждой группы крови соответственно.



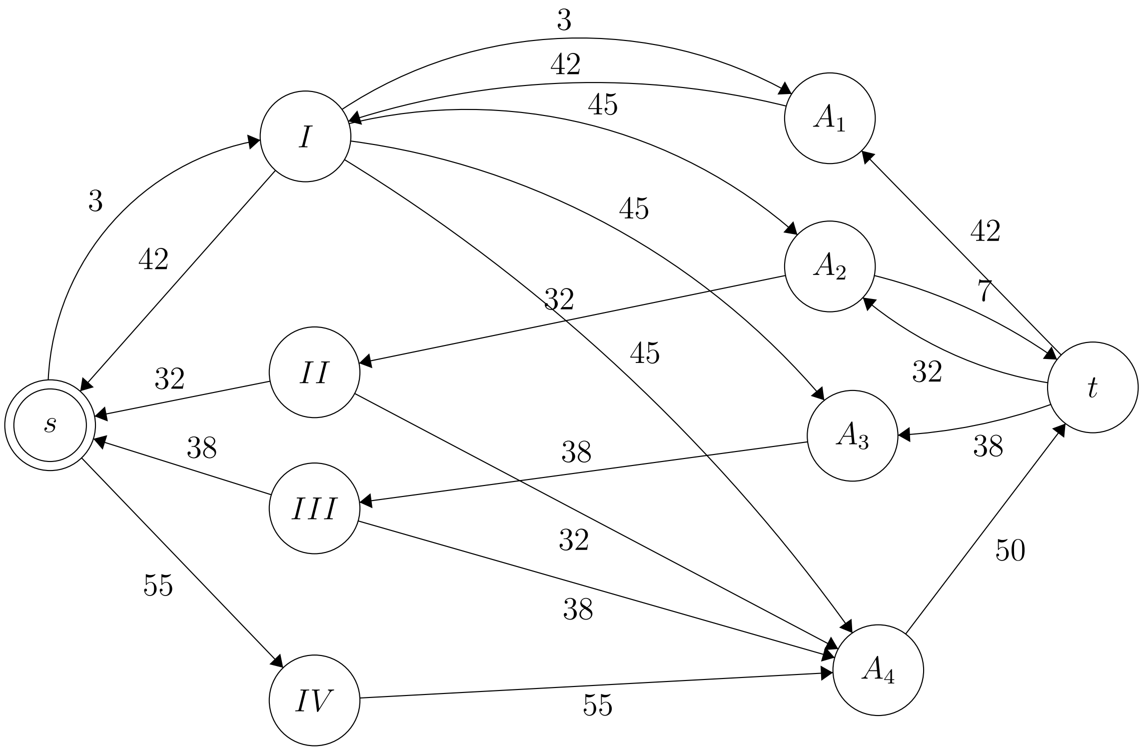
$1) s - I - A_1 - t, c_{min} = 42,$



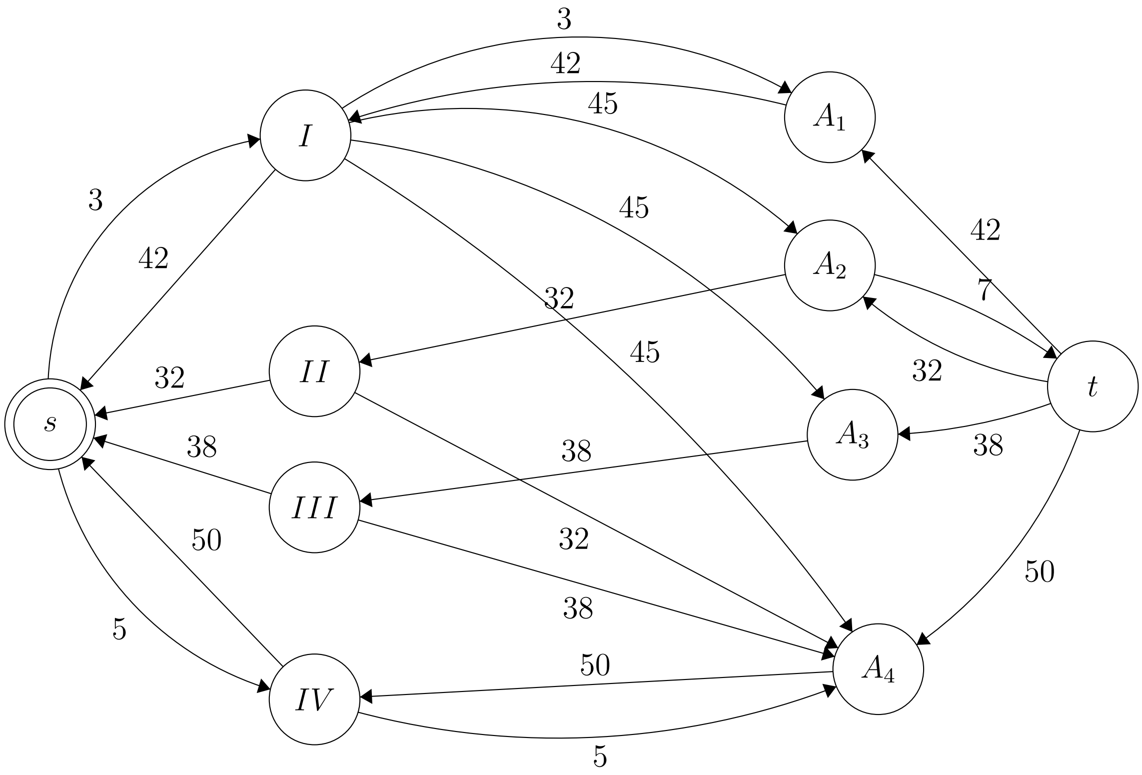
$2) s - II - A_2 - t, c_{min} = 32,$



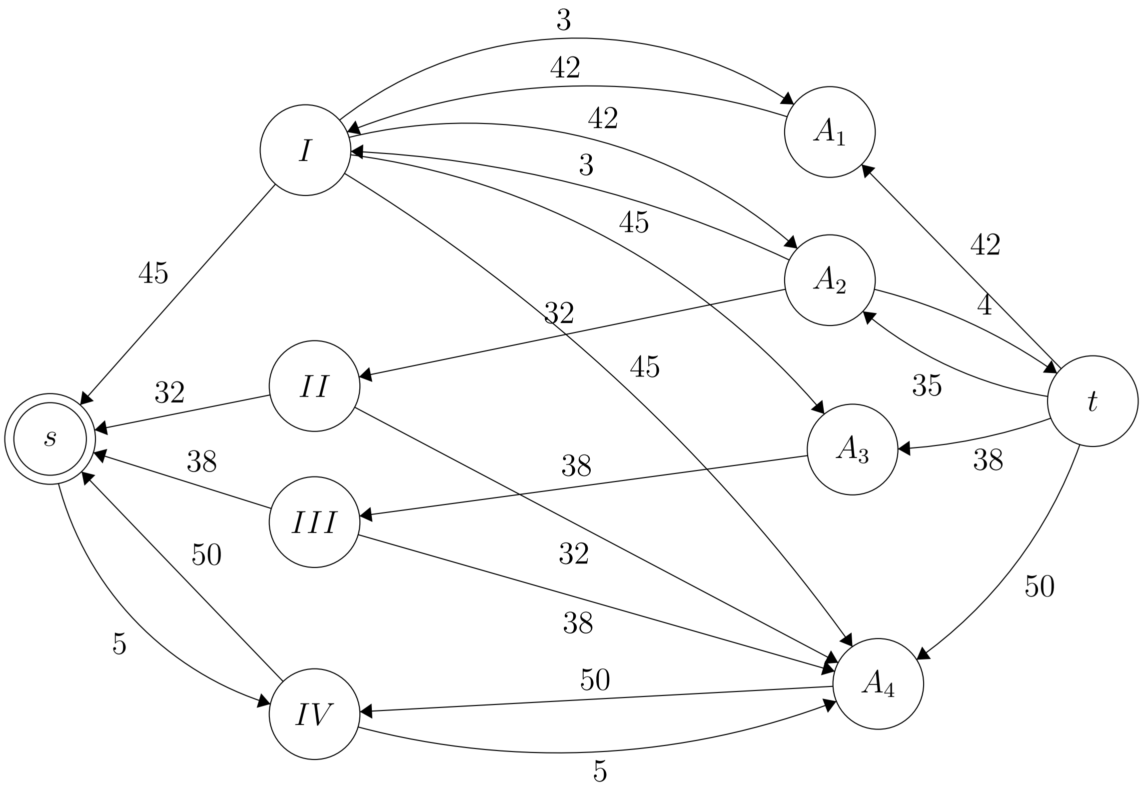
$3) s - III - A_3 - t, c_{min} = 38,$



$4) s - IV - A_4 - t, c_{min} = 50,$



5)$s - I - A_2 - t, c_{min} = 3$



t не достижима из s. Максимальный поток: $42 + 3 + 38 + 50 = 165$, необходимо 169.
(ii) Всех пациентов обслужить нельзя так как, чтобы обслужить пациентов с I и II группами крови необходимо 81 доза в сумме, а в наличии всего 77.

3

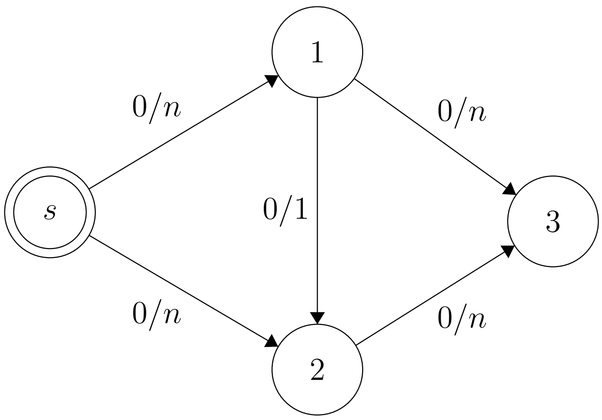
(i) Реберная $k-$связность.
Приведем полиномиальный алгоритм, который вычисляет максимальное k для данного графа. Пусть G — не ориентированный граф. Присвоим каждому ребру пропускную способность равную 1. Для каждой пары вершин (u, v) найдем максимальный поток из u в v. Граф является $k-$реберно связным

тогда и только тогда, когда максимальный поток любой вершины в любую не меньше k.
Корректность:
1) $\Longleftarrow$ Пусть граф является $k-$связным, докажем, что максимальный поток между любой парой вершин (u, v) не меньше k. Предположим противное: существует пара вершин u, v, между которыми максимальный поток меньше k. Тогда в минимальном разрезе меньше чем k ребер. Удаляем последовательно ребра. Получаем, что после удаления меньше чем k ребер, граф становится несвязным. Противоречие.
2) $\Longrightarrow$ Пусть максимальный поток между любой парой вершин не меньше k. Предпологая, что граф не является $k-$связным, получаем противоречие аналогично пункту 1).
Сложность:
находим максимальный поток с помощью алгоритма Эдмондса-Карпа за $O(|V||E|^2)$. Находим максимальный поток для каждой пары за $O(|V|^2) \cdot O(|V||E|^2) = O(|V|^3|E|^2)$.

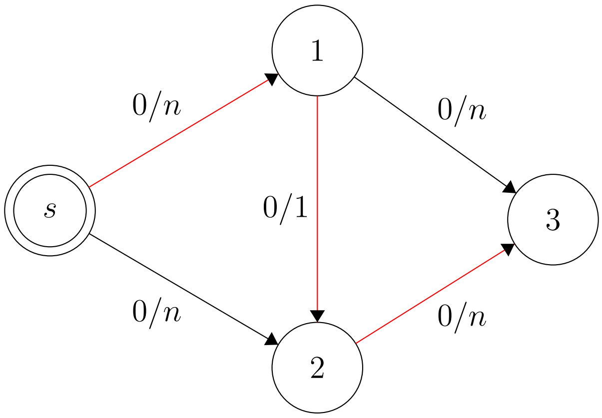
(ii) Вершинная $k-$связность.
Идея алгоритма: переберем все пары вершин (u, v) и найдем минимальное число вершин, которое надо удалить, чтобы разделить вершины u, v. Для этого перестроим граф G в граф G' следующим образом: для каждой вершины a создадим две ее копии a_1, a_2 , одну для входящих ребер, другую для выходящим, и соединим их ребром (a_1, a_2) с пропускной способностью 1. Далее действуем аналогично со случаем $k-$реберной связности.
Корректность:
количество ребер в минимальном разрезе равно максимальному потоку и равно количеству вершин, которые необходимо удалить, чтобы разделить u, v. Корректность следует на прямую из построение графа G' и коректности алгоритма нахождения максимального k, при котором граф остается реберно связным.
Сложность:
перестраиваем за $O(|V|)$, далее за $O(|V|^3|E|^2)$ находим максимальное k.

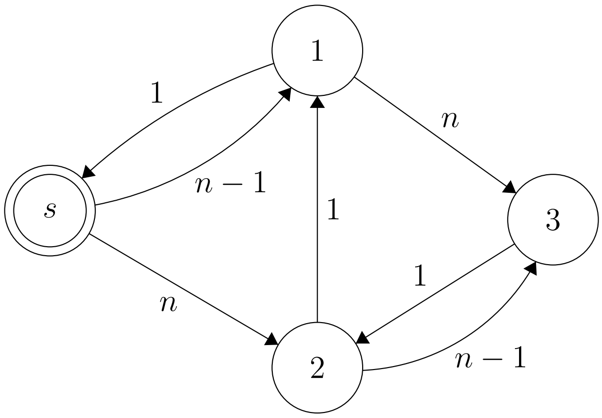
4

Алгоритм не всегда будет полиномиальным, потому что мы выбираем **любой** путь, а не кратчайший. Пример:



$n = 2^{|V|}$
Выберем на первой итерации алгоритма путь, обозначенный красным на рисунке ниже и построим остаточную сеть.





На следующей итерации веберем путь, опять включающий ребро $(2, 1)$. Плучает, чтобы получить граф, в котором вершина t не достижима из s, понадобится $\Theta(2^{|V|})$ итераций. Очевидно, алгоритм не является полиномиальным по длине входа (количеству вершин). Можно заметить, что с помощью BFS максимальный поток находится за 2 итерации.

5

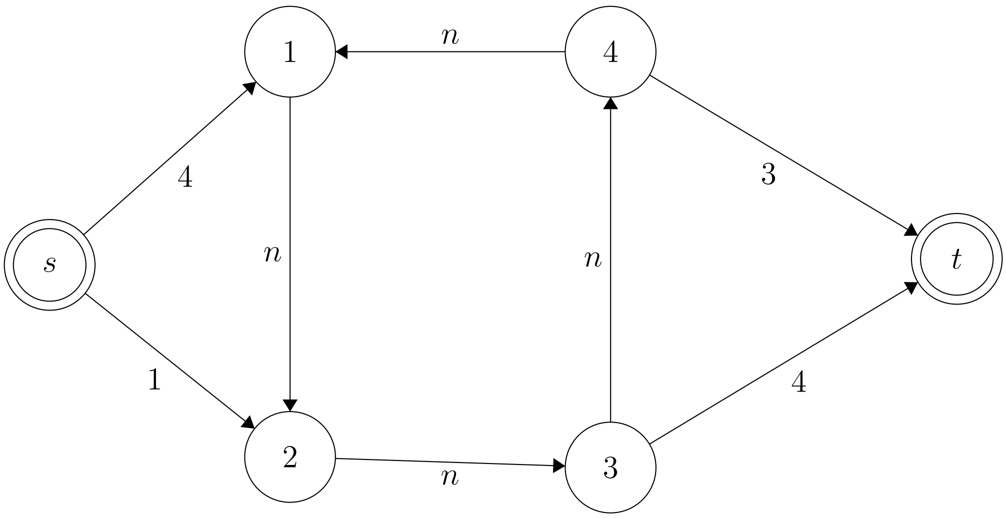
Если $f(u, v) = f(v, u)$, следовательно, все ребра ориентированы в обе стороны ($\longleftrightarrow$). Поступим также как в задаче $4(ii)$: для каждой вершины $v \in V$ создадим 2 копии этой вершины и направим в одну вершину все входящие ребра, а из второй все выходящие. Эту пару вершин соединим ребром с пропускной способностью $c(v)$.

6

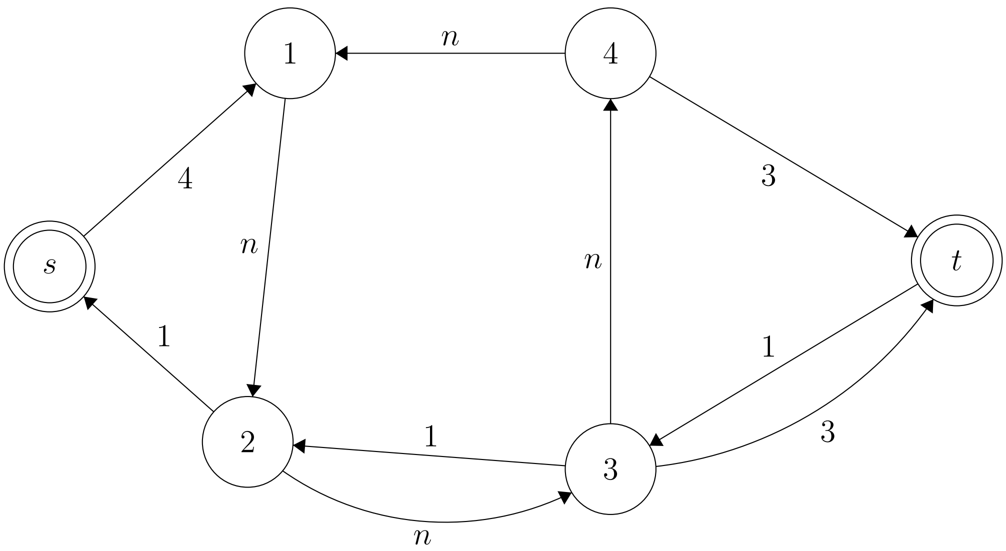
С помощью алгоритма Куна находим максимальное паросочетание в графе. Красим найденные ребра в первый цвет, затем удаляем эти ребра. Так делаем d раз, пока не удалим последние n ребер. Получаем граф, из каждой вершины которого исходят ребра разных цветов.
Корректность:
Докажем, что на каждой итерации алгоритма находим совершенное паросочетание (включающее все вершины графа). Лемма Холла утверждает, что если в двудольном графе для любого натурального k любые k вершин одной из долей связаны с по крайней мере с k вершинами другой, то граф разбивается на пары. Докажем, что на каждой итерации алгоритма выполняется условие леммы Холла. На первой итерации стпень каждой вершины равнаd, выберем произвольное подмножество $n \geq k$ вершин из первой доли. Предположим, что выбранные k вершин первой доли не связаны с k вершинами другой доли. Это означает, что степень каких-то вершин во второй доле больше d, что противоречит условию, следовательно, на перовй итерации в графе есть совершенное паросочетание. Так как мы на каждом шаге удаляем равно одно ребро между парой вершин, степень каждой за одну итерациию уменьшается на единицу. Следовательно, на каждой итерации алгоритм будет находить идеальное паросочетание.
Сложность:
На каждой из d итераций запускаем алгоритм Куна сложностью $O(|E||V|)$. Суммарная сложность: $O(|E||V|d)$.

7

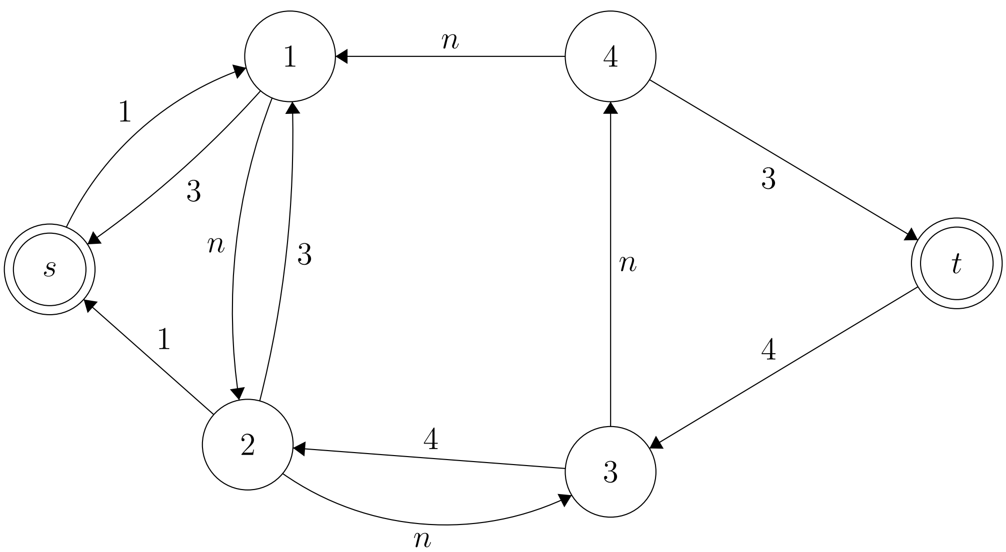
(i) Перестроим граф так, как написано в условии.



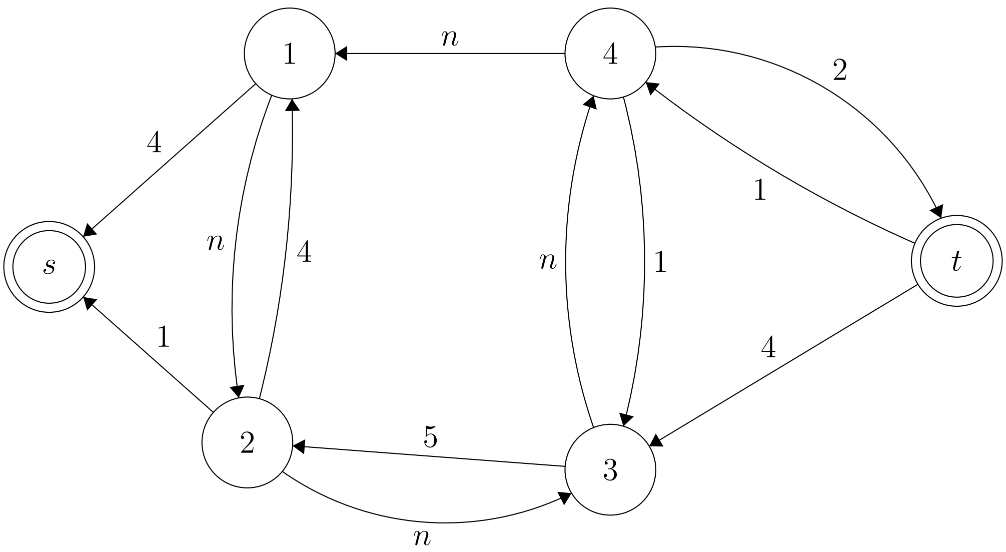
$n = \infty$
$1) s - 2 - 3 - t, c_{min} = 1,$



$2) s - 1 - 2 - 3 - t, c_{min} = 3,$



3)$s-1-2-3-4-t, c_{min}=1$,



t не достижима из s. Максимальный поток: $f_{max}=5$.
(ii) Найдем минимальный разрез. Минимальный разрез — это разрез, имеющий минимальную суммарную пропускную способность. Очевидно, что в минимальном разрезе не будет остаточных ребер (ребра, направленные из доли, содержащей вершину s, в долю, содержащую вершину t) . Следовательно, минимальный разрез для графа, изображенного выше: $(1,s)-(2,s)$.
(iii) Данным примером проиллюстрировали теорему Форда-Фалкерсона: величина максимального потока в графе путей равна величине пропускной способности его минимального разреза.
Доказательство:
Очевидно, что поток не больше чем пропускная способность просто исходя из определений.
Докажем, что не меньше. Предположим, пропускная способность минимального разреза меньше чем поток. Если рассматриваемый разрез минимальный, значит, во всех остальных разрезах, пропускная способность которых больше, есть остаточные ребра. А это же значит, что вершина t будет достижима из s, следовательно, существует увеличивающий путь, следовательно, поток не максимален, получаем противоречие. Теорема доказана.