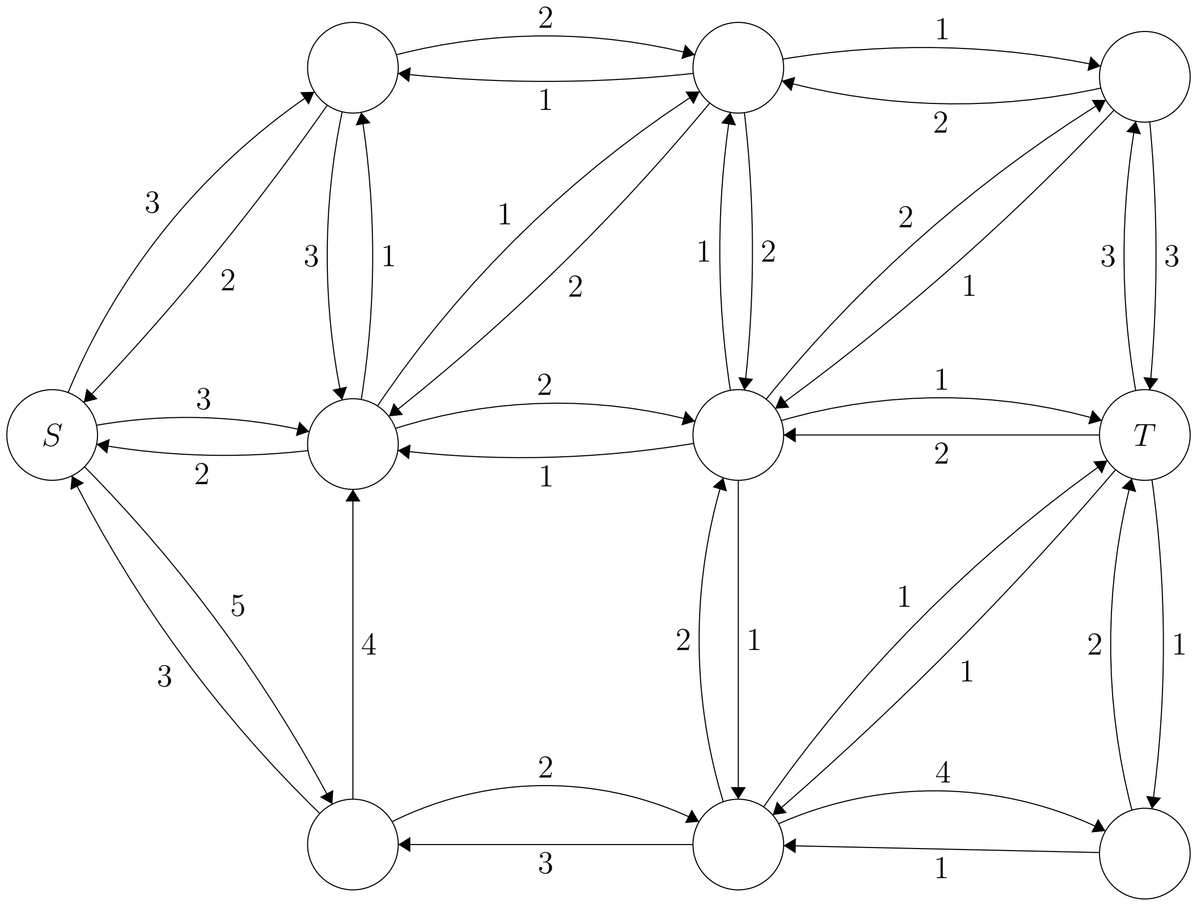
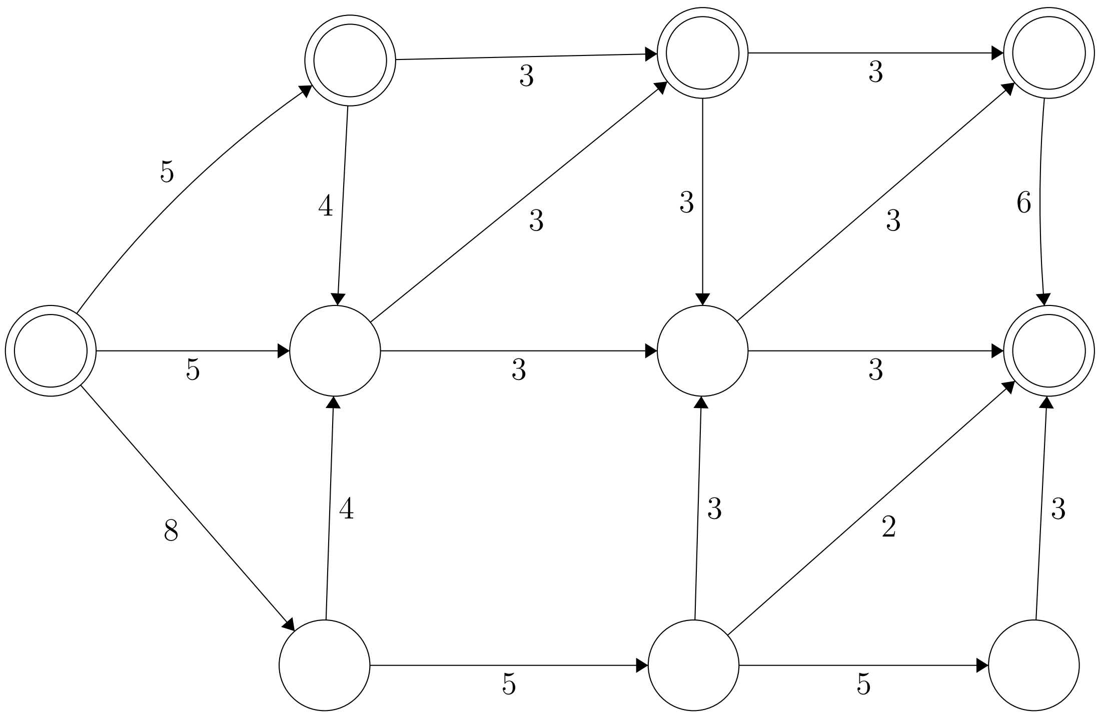
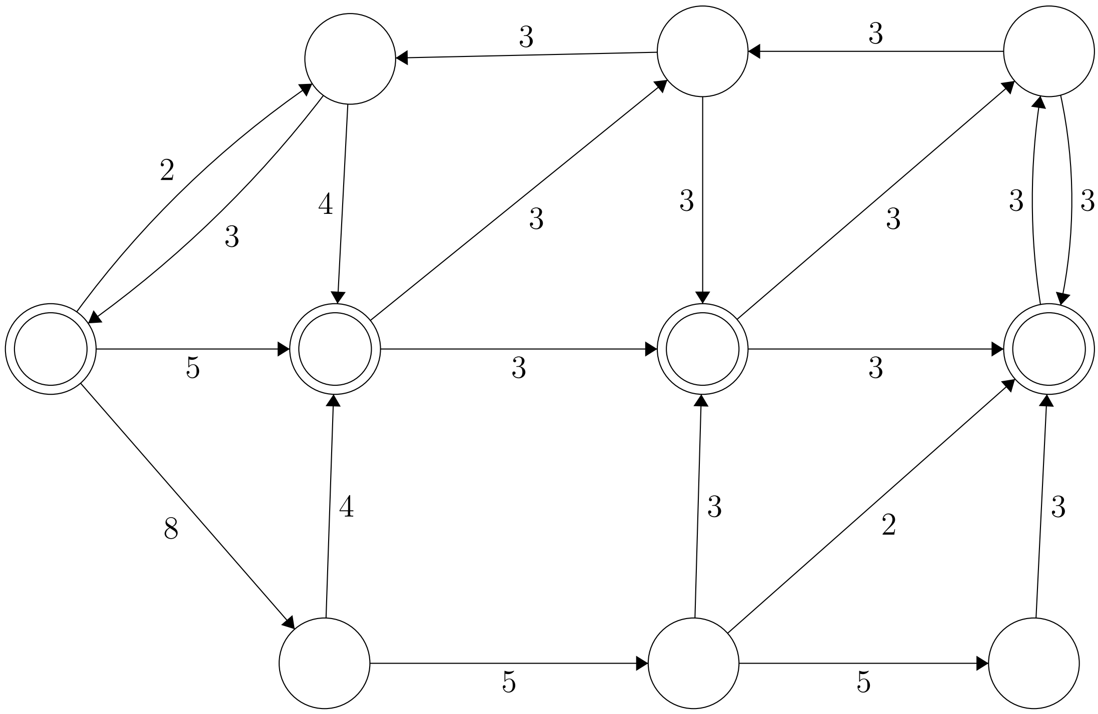
Задача 1

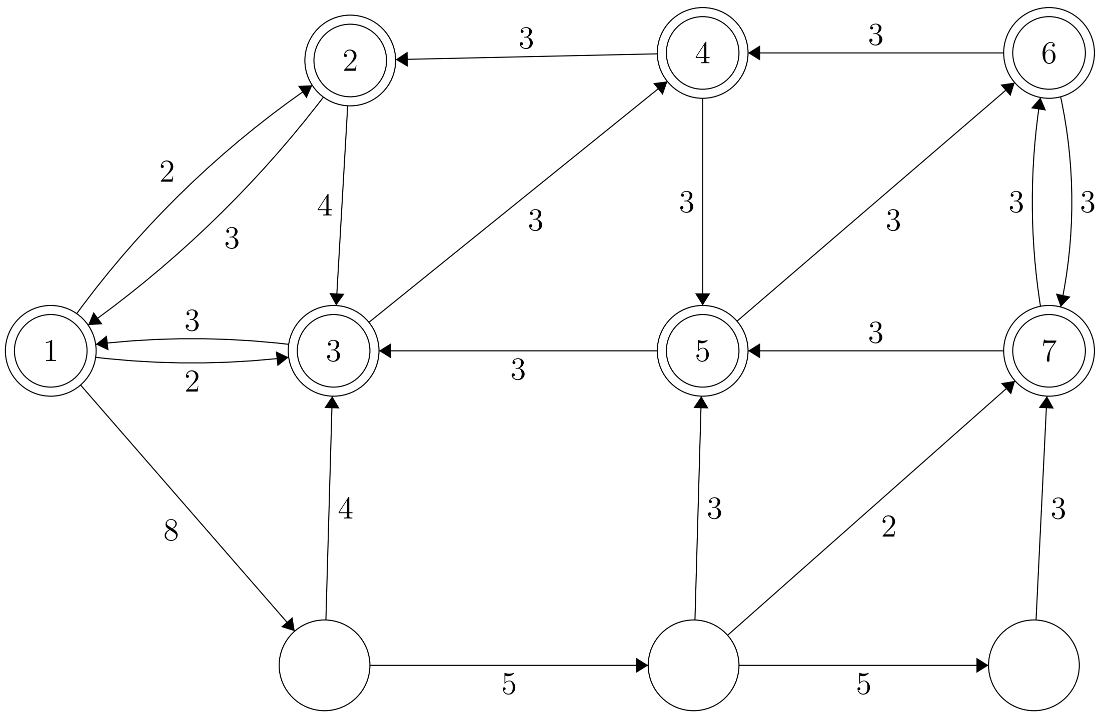
(i) Поток равен $f = 7$
(ii)



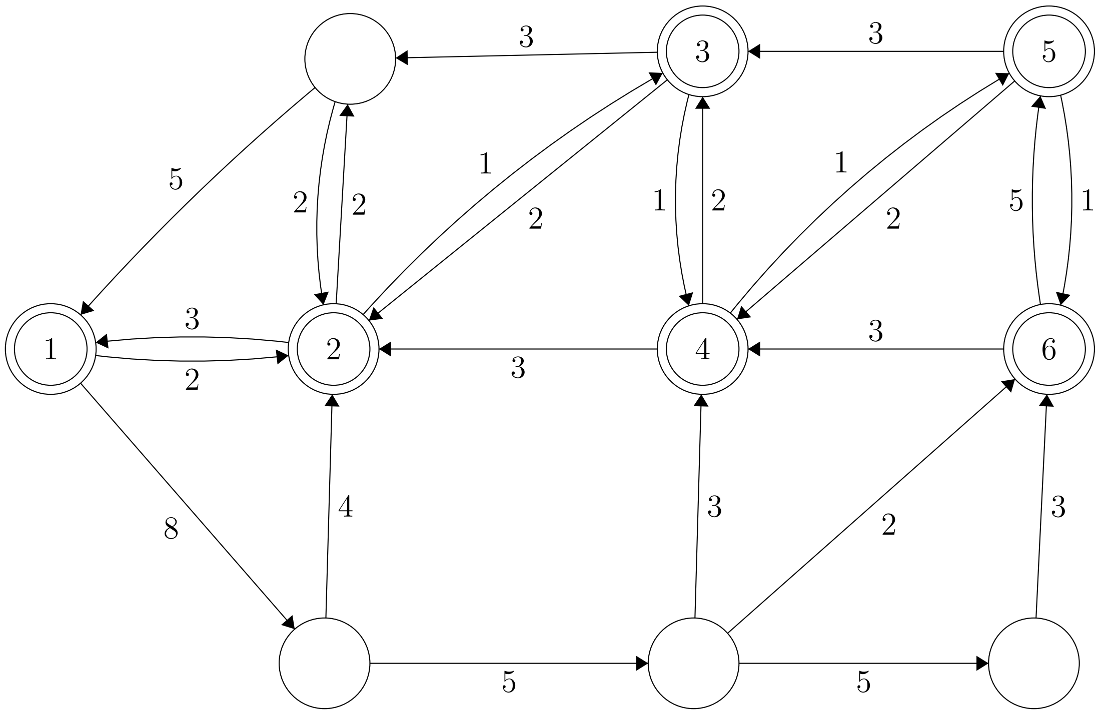
(iii) Поток f не максимален, поскольку он меньше максимального потока, найденного в следующем пункте и равного 140. Также в данной остаточной сети есть увеличиающий пути из s в t.
(iv) Вершины увеличивающего пути отмечены (кружками и номерами обхода вершин).


5
3
3
4
3
3
3
6
5
3
3
8
4
3
2
3
5
5

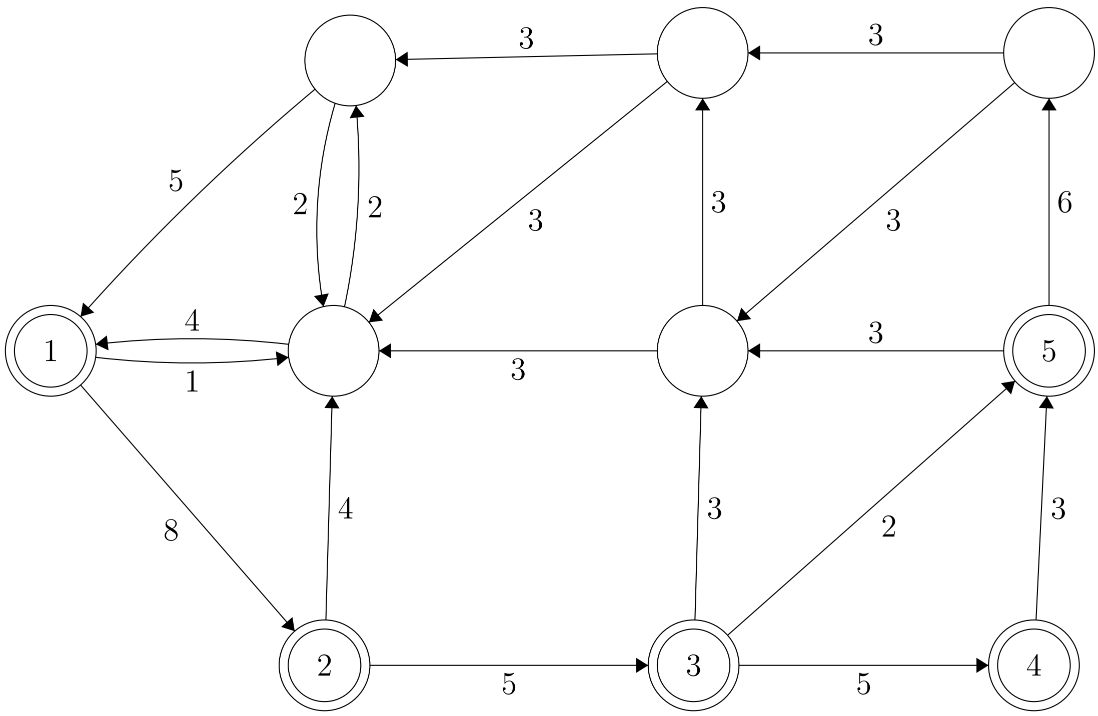

3
3
2
4
3
3
3
3
3
3
5
3
3
8
4
3
2
3
5
5

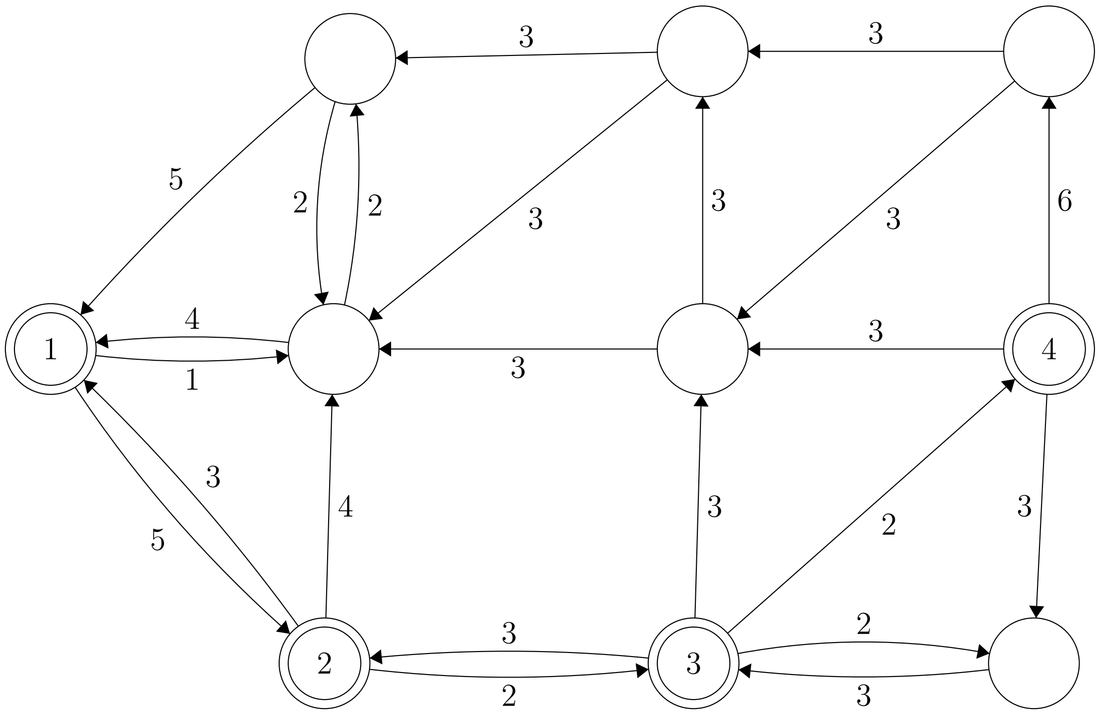


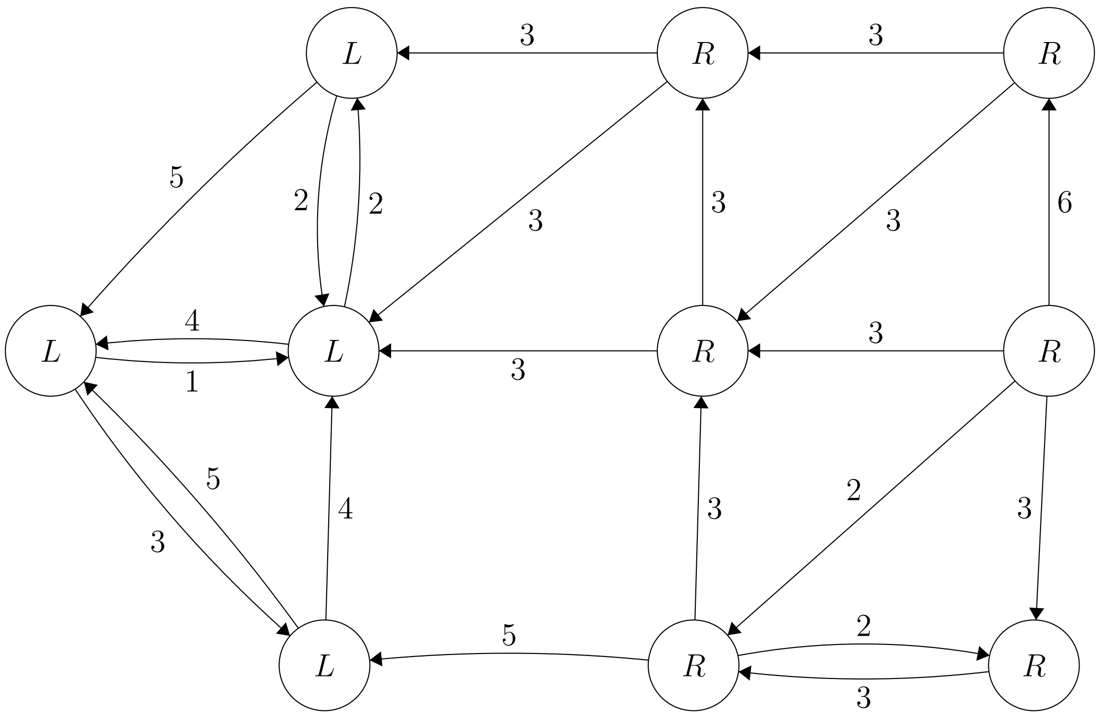
2
4
6
3
3
2
4
3
3
3
3
3
1
3
3
5
3
7
2
3
3
3
8
4
3
2
3
5
5



3
3
5
3
5
1
5
2
2
2
1
2
1
2
1
2
5
1
3
1
2
4
3
6
2
3
3
8
4
3
2
3
5
5


3
3
5
2
2
3
3
3
6
1
4
1
3
5
3
8
4
3
2
3
2
5
3
5
4


3
3
5
2
2
3
3
3
6
1
4
1
3
4
3
3
5
4
3
2
3
2
3
2
3

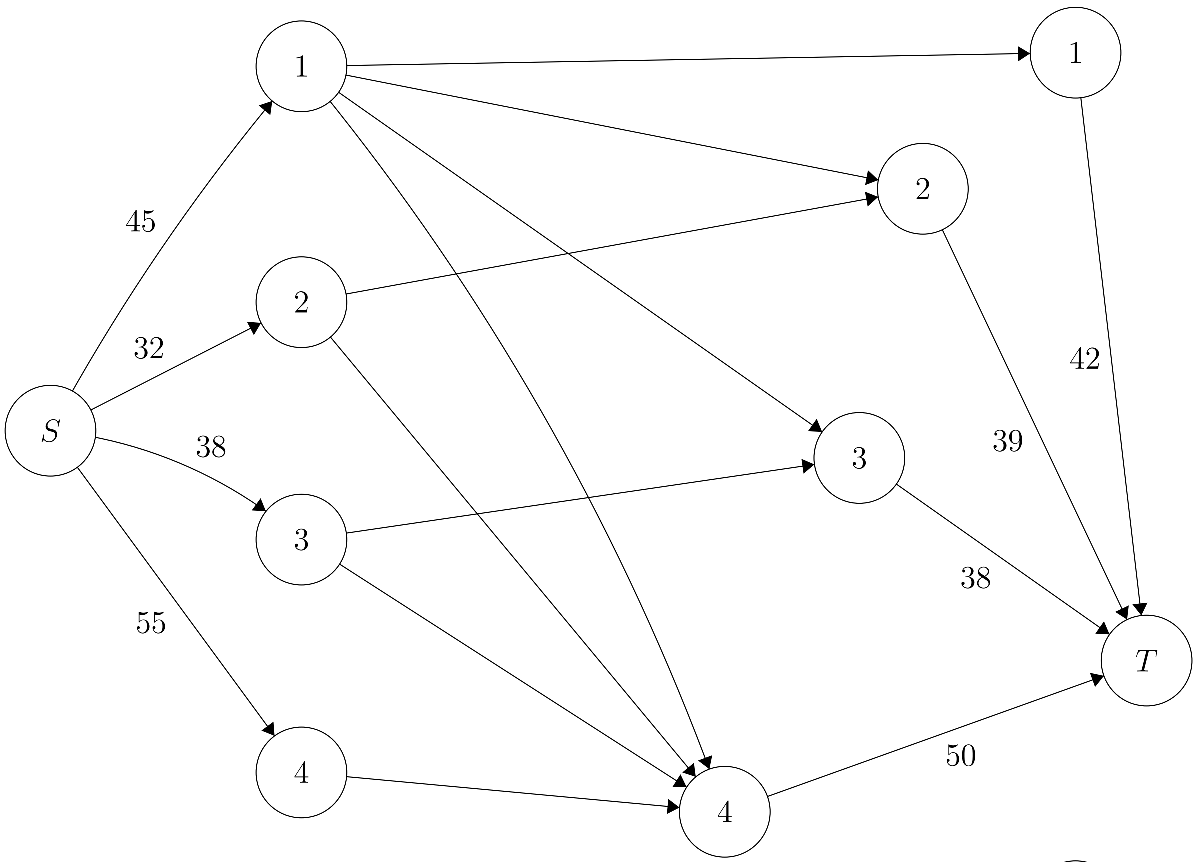


На последнем шаге вершина t недостижима из s. Максимальный поток равен пропускной способности полученного разреза в исходном графе (вершины разреза отмечены на последнем шаге). Получаем Max-flow = 14.

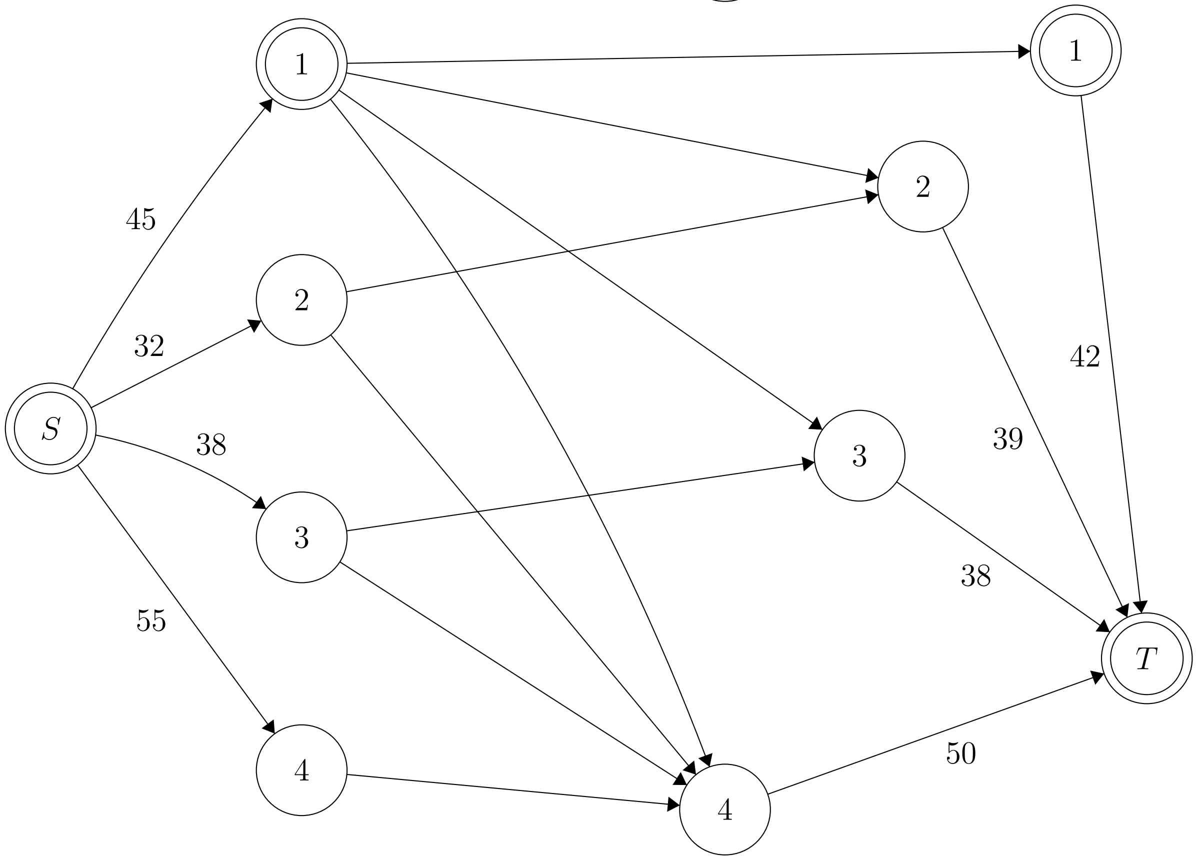
(v) Минимальным является разрез, указанный на последнем рисунке отметками вершин. Для его нахождения после работы обычного алгоритма Форда-Фалкерсона запустим BFS из истока. Достижимые из истока вершины положим в первую долю, остальные вершины попадут во вторую долю, содержащую сток. Пропускная способность минимального разреза равна максимальному потоку и равна 14.

Задача 2

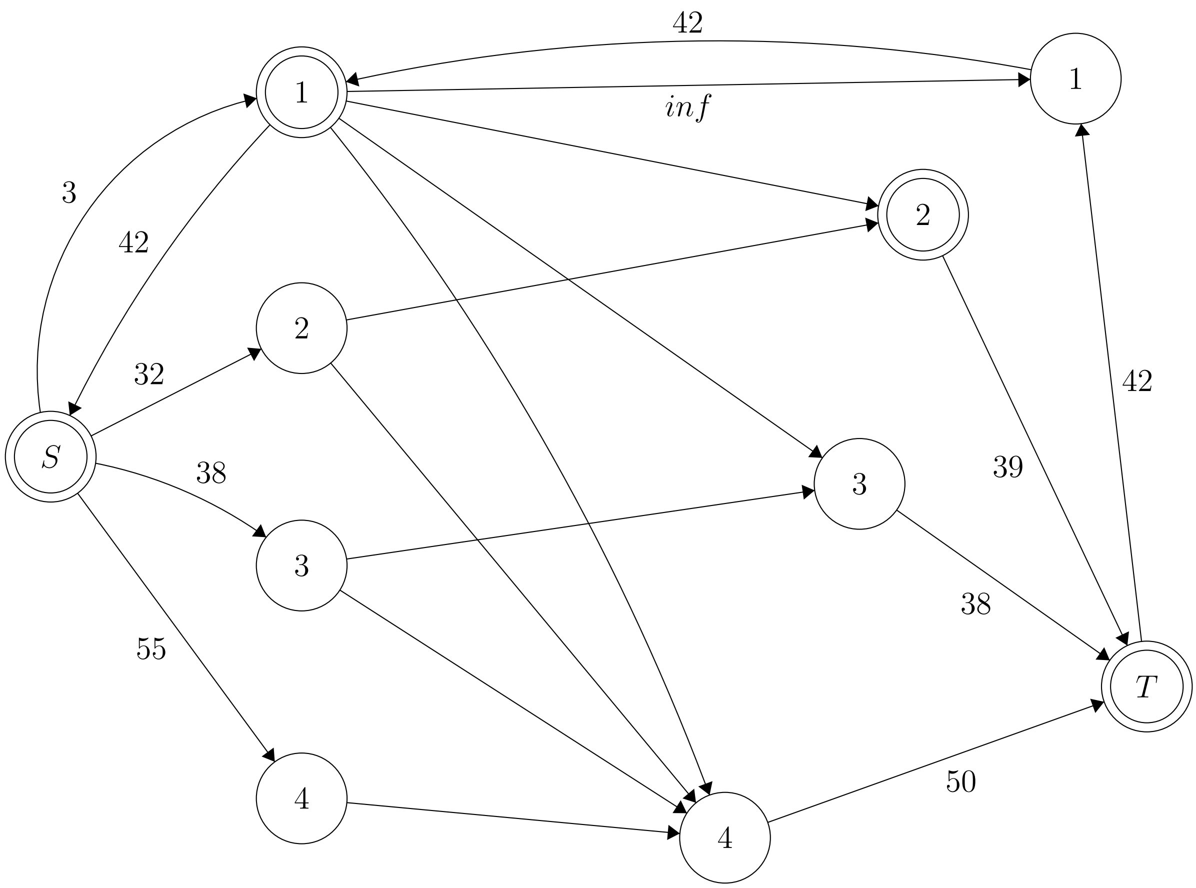
В следующем графе, где ребра, соединяющие цифры, имеют достаточно большую пропускную способность inf, найдем max поток из истока в сток.
Первый столбец цифр обозначает наличие групп крови, второй — запросы групп крови.

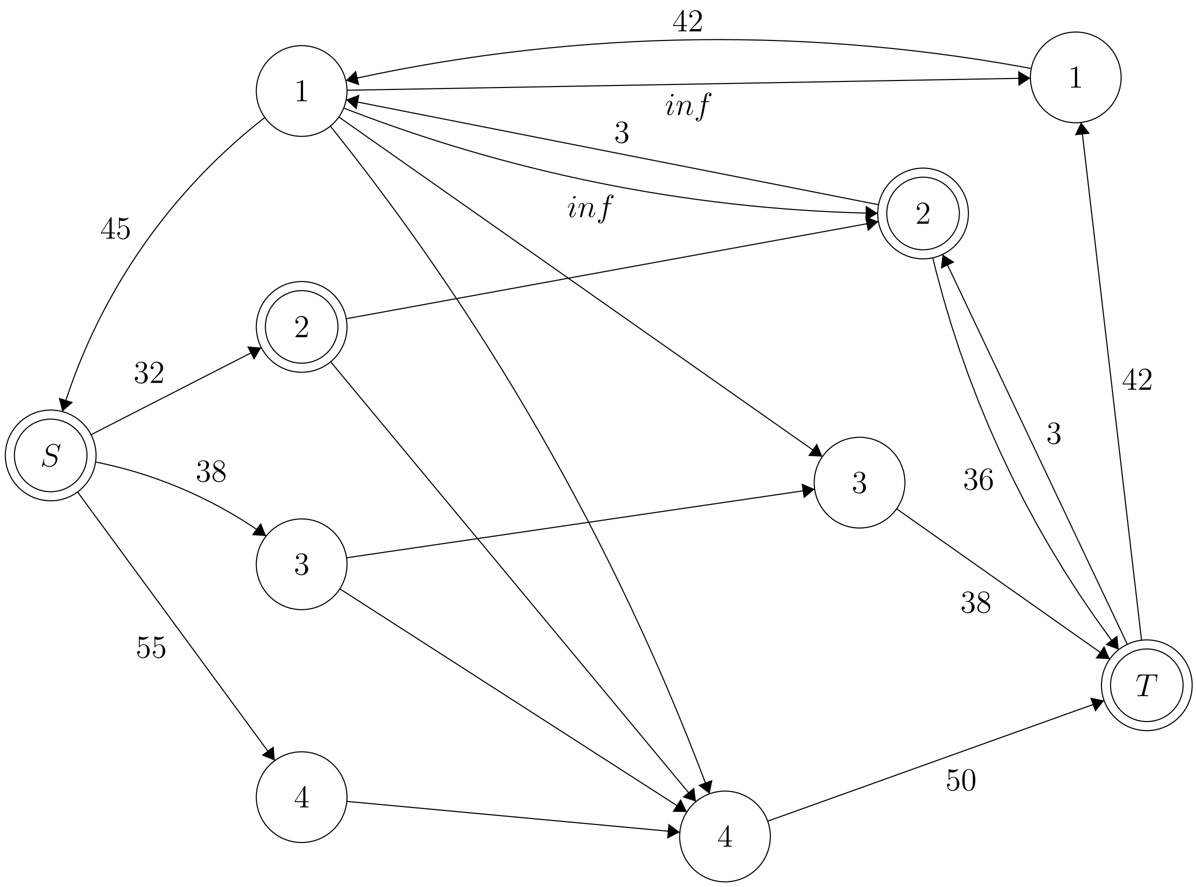


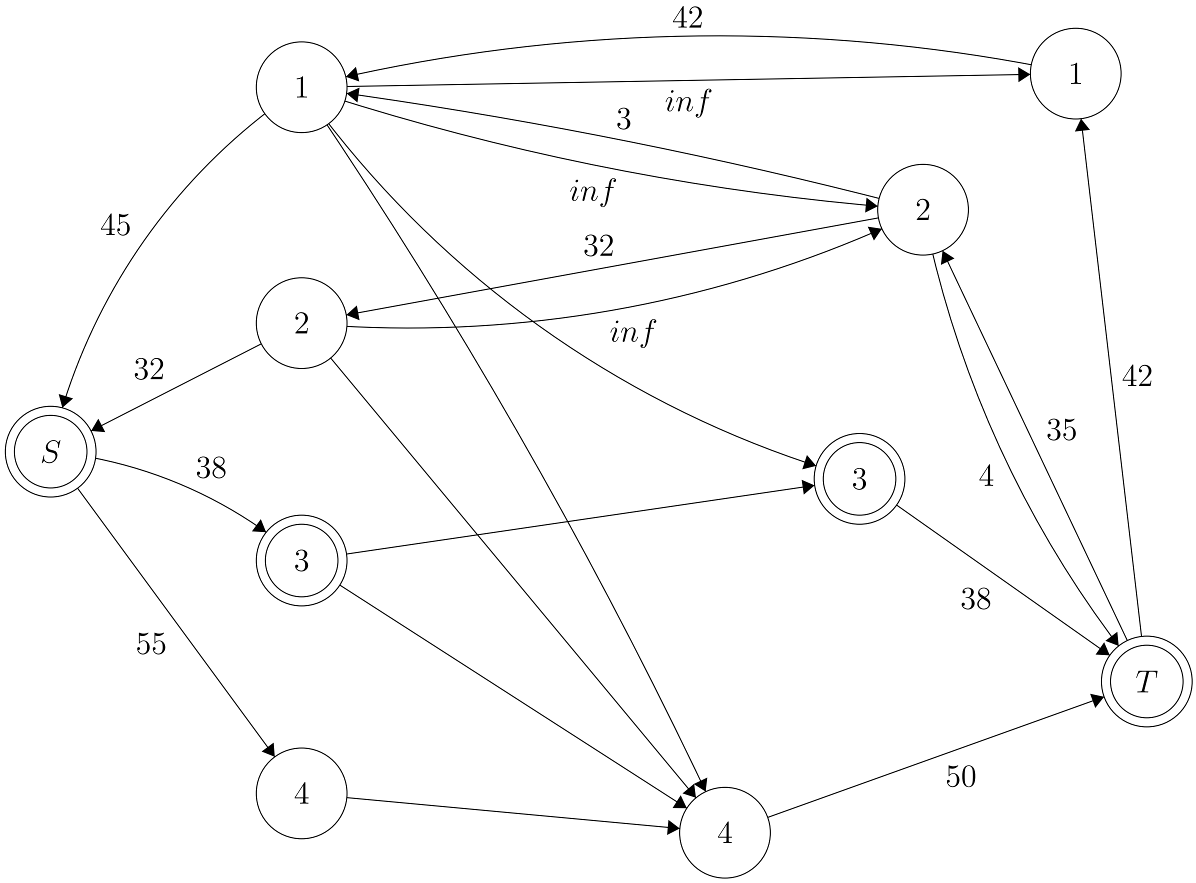
1
1
45
2
2
32
42
S
3
39
38
3
38
55
T
50
4
4

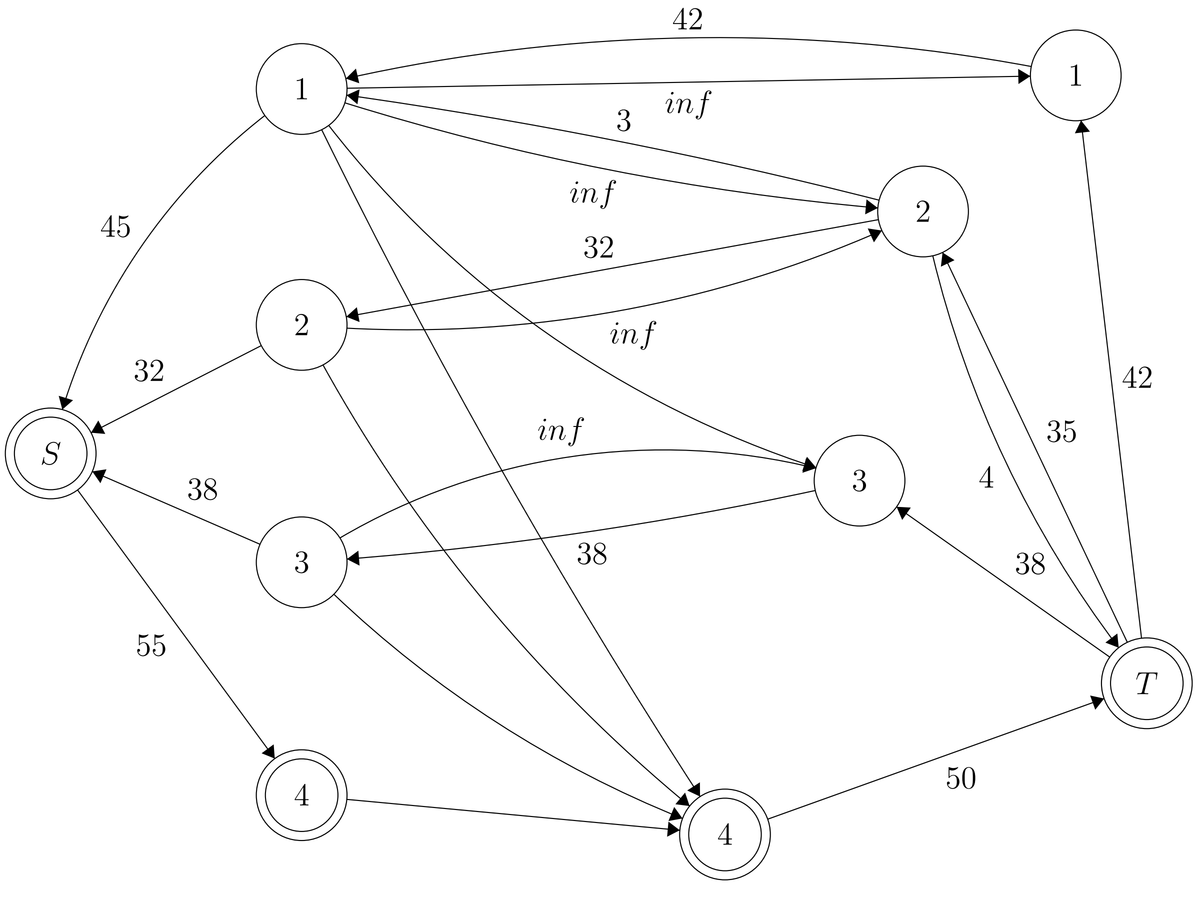


1
1
45
2
2
32
42
S
3
39
38
3
38
55
T
50
4
4

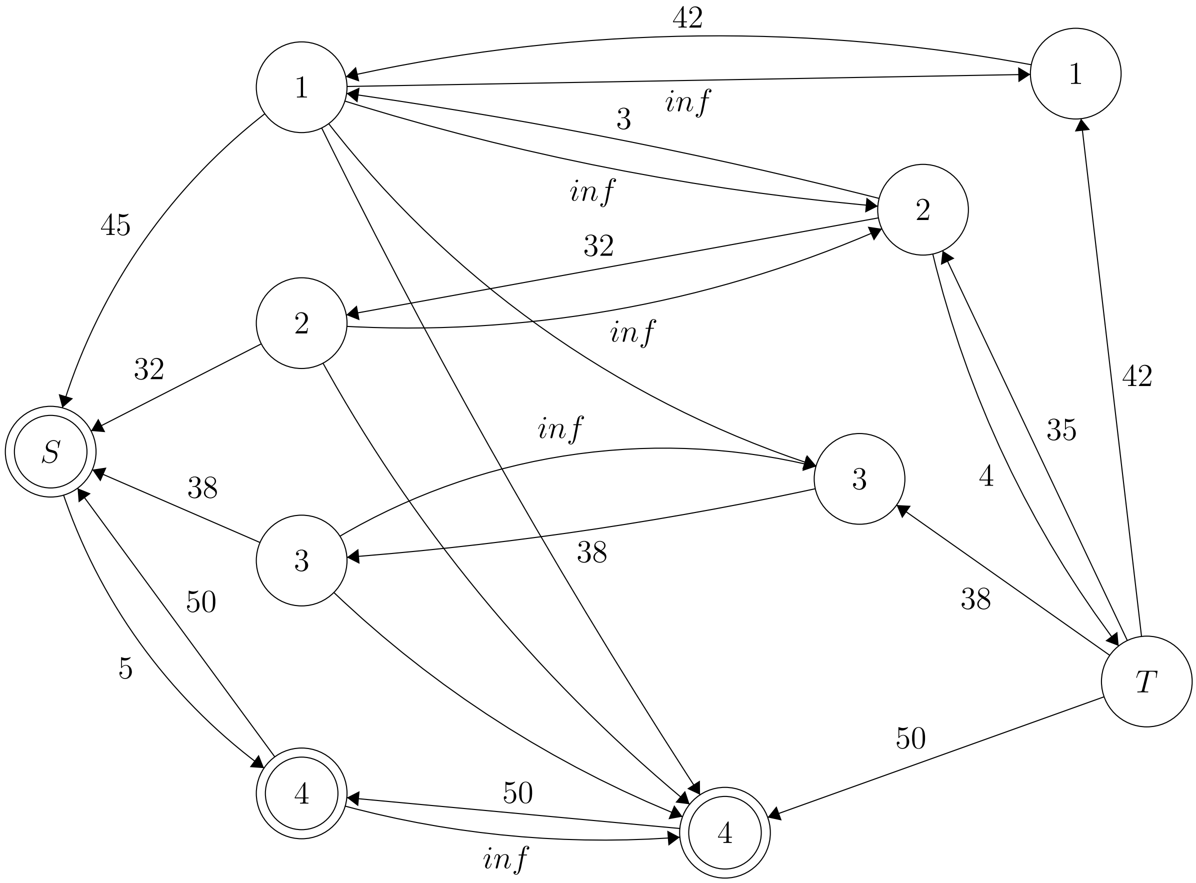

42
1
1
inf
3
42
2
2
32
S
38
3
3
39
42
55
38
T
4
4
50


42
1
1
inf
3
inf
2
45
2
32
S
38
3
3
36
3
42
55
38
T
4
4
50





До этого рисунка включительно выделенные вершины есть увеличивающий путь.



На последнем рисунке выделены вершины из одной доли минимального разреза, а невыделенные вершины принадлежат другой доле. Пропускная способность этого разреза есть max поток сети. Он равен 165.

(i) Т. е. можем обслужить максимум 165 пациентов, например, по следующему распределению:
всем людям с 1 группой дадим кровь 1 группы (42 дозы), людям со второй группой дадим всю кровь второй группы и еще 3 дозы первой группы (всего 35 доз), всем людям с 3 группой крови дадим по дозе 3 группы крови и аналогично для 4 группы.

(ii) Простое интуитивное решение задачи показывает невозможность раздать кровь всем: 4 группа подойдет только людям с 4 группой, поэтому людей с 4 группой обеспечим кровью их группы. Людям с третьей группой тоже дадим кровь из их группы. Для первыз двух групп есть запас из 77 доз, но нужно 81, поэтому обеспечить всех нельзя.

Задача 3

(i) Решим данную задачу за полиномиальное время. Для этого построим новый граф, в котором каждое ребро из исходного графа заменим на два ориентированных антипараллельных ребра веса 1 между соответствующими вершинами. Тогда если исходный граф k-связный, то в новом графе для любой пары вершин максимальный поток из первой вершины во вторую больше либо равен k и обязательно

есть разрезы с пропускной способностью равной k, чтобы связность нарушалась при удалении ребер из такого разреза. Пропускная способность не меньше k, поскольку иначе можно убрать меньше k ребер и нарушить связность графа.
Получение нового графа осуществляется линейно от числа вершин (а значит и от записи исходного графа), далее запускается полиномиальный от входа алгоритм Эдмонда - Карпа с асимптотикой $O(|V| \cdot |E|^2)$. Далее считаем количество ребер в минимальном разрезе и сравниваем его с числом k. Если для каждой пары вершин (их $|V|^2$) данные числа больше либо равны k и есть равные k, то ответ «да», иначе ответ нет. Получен полиномиальный алгоритм проверки.

(ii) Решим данную задачу за полином, используя теорему $Mengera$. По исходному графу построим новый граф раздвоением каждой вершины, при котором входящие в вершину ребра входят в одну из новых вершин, а выходящие из нее ребра выходят из другой новой вершины. По теореме реберная k-связность нового графа равносильна вершинной k-связности полученного графа. Тогда можем применить предыдущий пункт после данного линейного от числа вершин построения нового графа.

Задача 4

Рассмотрим худший случай выбора пути в алгоритме Форда - Фалкерсона, описанный тут $not-polynomial$. Вместо веса 1000 может быть число w. Такой алгоритм выполнит $O(w)$ итераций, что не является полиномиальным от задания графа длины $O(\log w)$.

Задача 5

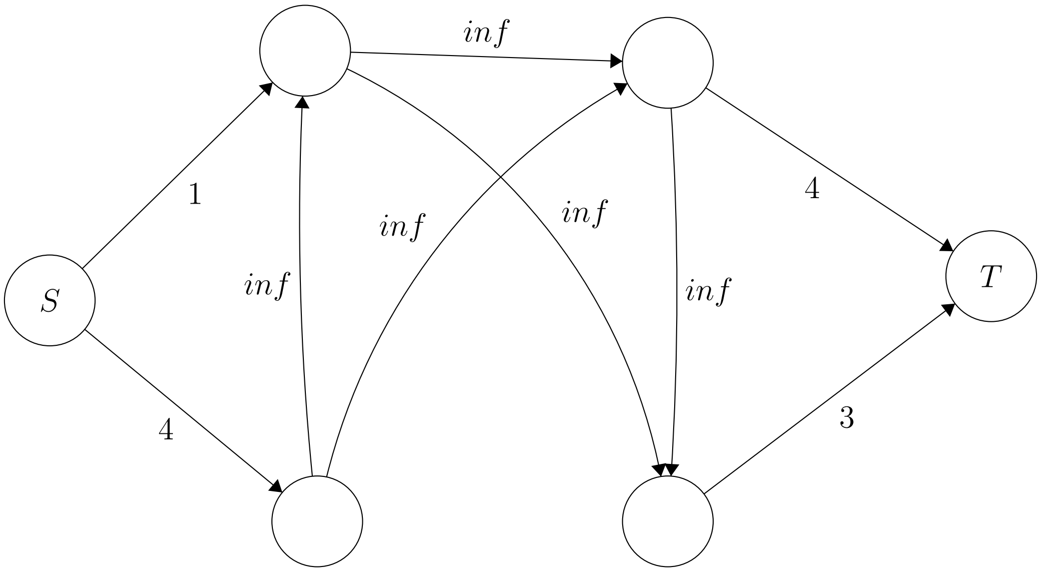
Повторим преобразование исходного графа из пункта 3(ii). Вес вершины будет весом соответствующего ребра, а все остальные ребра будут без ограничений на вес. В этом графе задача максимального потока равносильна таковой в исходной графе. Численные ответы для этих двух задач равны.

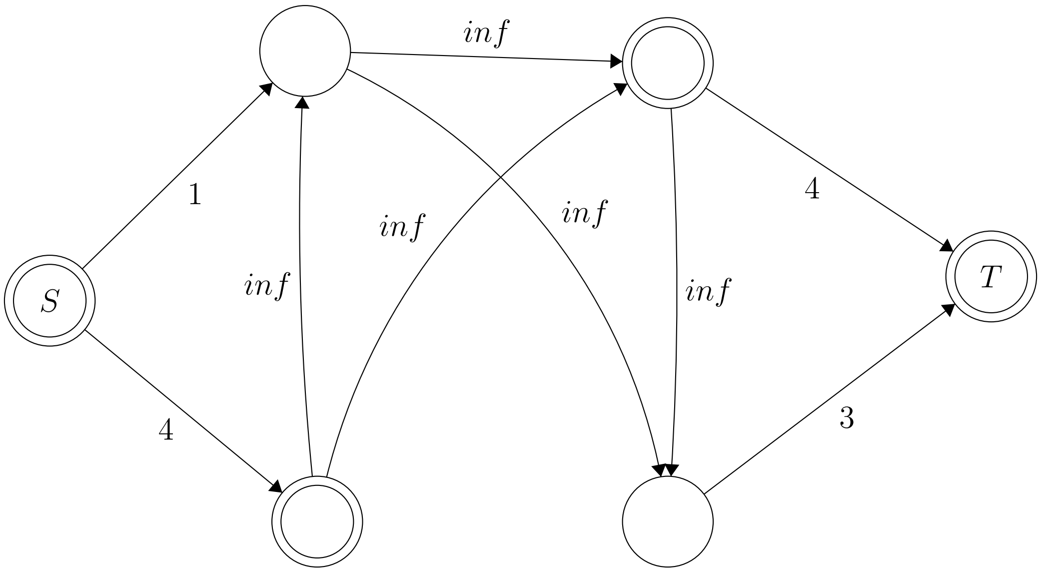
Задача 6

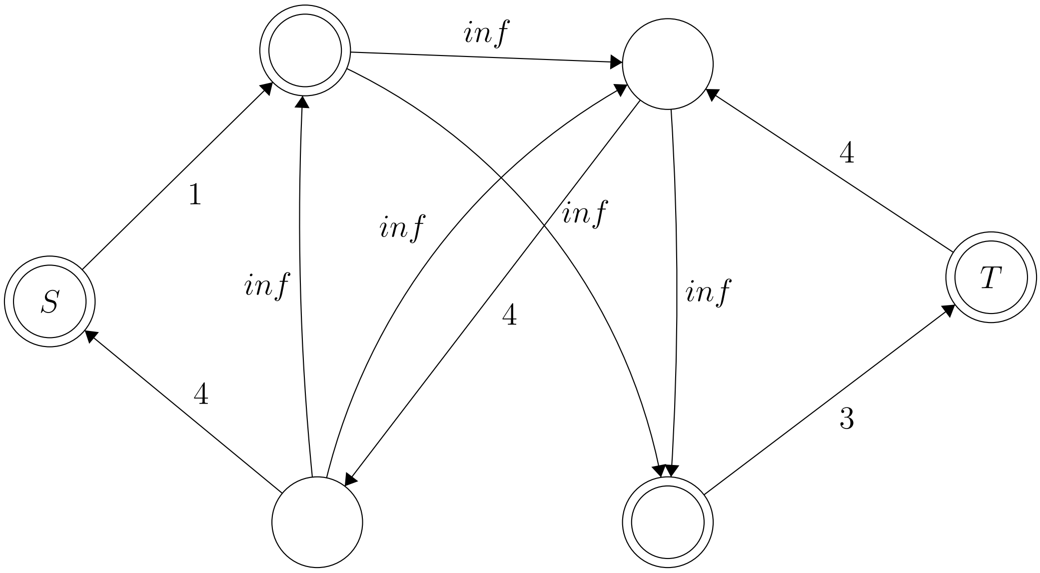
На каждом шаге из указания выполняется лемма Холла: для множества A вершин одной доли будет $|A|$ вершин во второй доле, что следует из того, что из каждой вершины выходят и входят по d ребер в другую и из другой доли соответственно. Поскольку $|A| \leq N(A) = |A|$, то лемма Холла верна и можем найти и покрасить в один цвет совершенное паросочетание. После того, как оно будет удалено, степени вершин уменьшатся на один. Значит, по аналогичным рассуждениям, можно выделить совершенное паросочетание и так на каждом шаге. В итоге, чтобы раскрасить двудольный граф из условия, нужно d раз найти совершенное паросочетание, что сделаем за $O(d \cdot |V|^3)$ в худшем случае (при реализации из $e - maxx$)

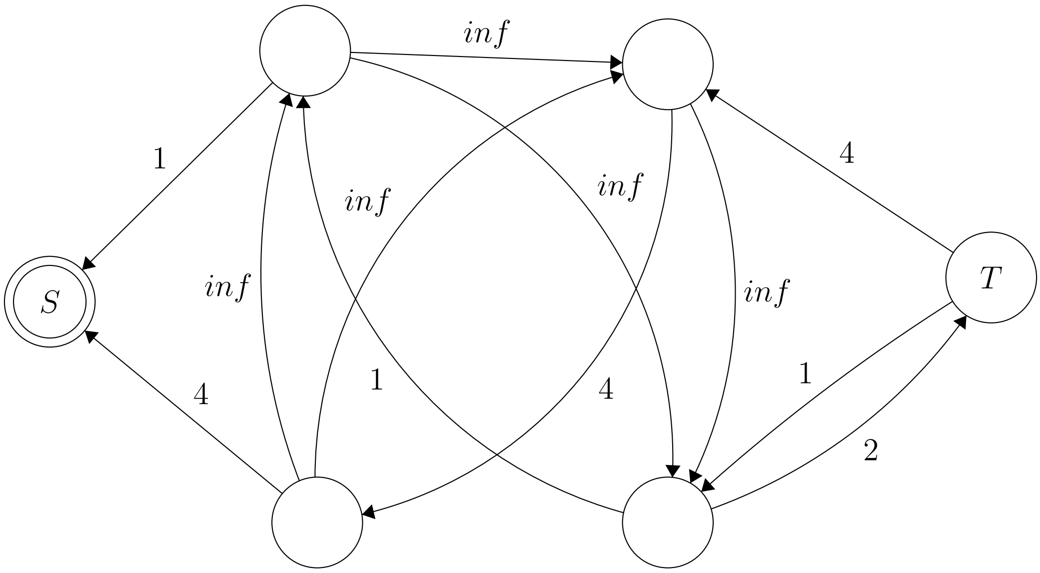
Задача 7

(i)









Максимальный поток равен 5 (это пропускная способность минимального разреза из пункта (ii))
(ii) Получен min разрез: в одной доле исток, в другой все остальные вершины.

(iii) В общем случае, доля разреза со стоком есть нужное множество. Нужно максимизировать сумму положительных весов проектов в доле с истоком и отрицательных весов проектов в доле со стоком. Эта сумма равна разности суммы весов всех проектов в доле с истоком и пропускной способности разреза с такими долями. Если пропусная способность разреза минимальна, то требуемая сумма максимальна, поэтому способ работает верно.