Максимальные паросочетания

Содержание

1 Задача про замощение фигуры доминошками

Из клеточного листа бумаги размера $N * N$ вырезали некоторую фигуру. Замостите её доминошками, а если её невозможно замостить целиком, то положите максимальное число доминошек. Доминошки — это плитки размера $2 * 1$; каждая доминошка занимает две соседние клетки. Идея решения: Рассмотрим различные примеры входных данных. На рисунке 1 приведено три фигурки из клетчатой бумаги. На первую фигурку можно положить без перекрытия только две доминошки. Возможное положение доминошек обозначено чёрточками. На вторую фигуру из клеток можно положить пять доминошек, при этом останется две незакрытые доминошками клетки, на третью — одиннадцать. Понятно, что число доминошек меньше либо равно числу клеток в фигуре пополам.

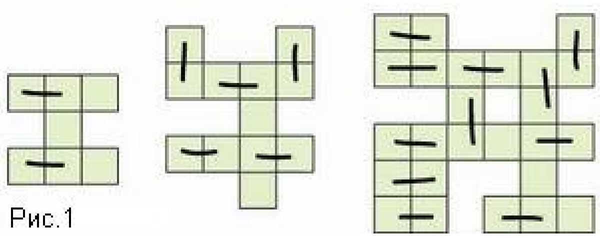


Рис.1

Один из возможных алгоритмов замощения фигуры доминошками может быть таким: находим пару соседних непокрытых клеток и кладем на них сверху доминошку; затем снова ищем пару соседних непокрытых клеток и кладём на них доминошку и так далее, пока не получим ситуацию, в которой нет соседних непокрытых клеток.

Задача Приведите пример, когда такой алгоритм находит неоптимальное решение, то есть располагает на фигуре меньше доминошек, чем в принципе возможно.

Такой алгоритм называется “жадным”. Жадные алгоритмы на каждом шаге ищут улучшение ситуации (например, первое попавшееся). Иногда жадным алгоритмам удается найти оптимальное решение. Но для указанного алгоритма это не так. Идея точного решения этой задачи основана на методе раскраски. Представим себе, что клетки бумаги раскрашены в чёрный и белый цвет, как клетки шахматной доски. Заметим, что доминошка покрывает одну чёрную и одну белую клетку. Будем говорить, что чёрная и белая клетка спарены, если на них лежит доминошка. Каждая клетка может быть спарена с одной из соседних клеток противоположного цвета. Можем подсчитать число чёрных и белых клеток. Очевидно, что количество доминошек не может быть больше, чем минимум из этих двух чисел. Для всех трёх случаев, изображённых на рисунке 1, данная верхняя оценка в точности совпадает с ответом — максимальным числом доминошек.

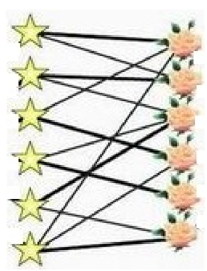
Задача Приведите пример связной клетчатой фигуры, для которой максимально возможное число доминошек меньше минимума из числа белых и чёрных клеток.

Как написать программу, которая всегда находит оптимальное решение? В качестве входа укажем программе размер листа N, а затем в каждой из N строк – допустимые и запрещённые поля. Например, знак ’#’ отмечает клетки которые вырезаны, а точки - клетки фигуры:

```
Вход №1          Вход №2
3                4
...              ....
...              #.##
...              ...#
                 #.##
Выход №1         Выход №2
4                3
```

2 Максимальное паросочетание

Это задача про девочек и мальчиков. Пусть у нас есть M мальчиков и D девочек. Каждый мальчик может дружить с несколькими девочками, и каждая девочка может дружить с несколькими мальчиками. Известно, кто с кем дружит. Нужно составить как можно больше пар мальчик–девочка, в которых мальчик и девочка дружат. Каждый ребенок может быть только в одной паре. При этом некоторые дети могут остаться без пары.



Обозначим девочек цветочками, а мальчиков – звездочками. Цветочек будем соединять линией со звездочкой, если соответствующие девочка и мальчик дружат. Мы получим то, что в дискретной математике принято называть двудольным графом. Некоторые из этих линий нам нужно сделать жирными, чтобы отметить, что соответствующую пару мы выбрали. Из каждой вершины может исходить только одна жирная линия (каждый ребенок может быть только в одной паре). Нужно получить как можно больше жирных линий.

2.1 Жадный алгоритм

Создаём пары как попало, пока возможно.

Задача Придумайте граф дружб, для которого жадный алгоритм образования пар дает неоптимальное решение.

2.2 Алгоритм Влада Генкина – Миши Кумскова

Выбираем одного из самых “несчастных” людей, у которых минимально число возможных пар. В качестве пары выбираем самого “несчастного” из его партнёров. Если “несчастных” партнёров несколько, выбираем любого из них.

Этот лучше, чем в жадном алгоритме, поскольку люди, которым может не найтись пары, обслуживаются в первую очередь.

Задача Придумать граф дружб, для которого алгоритм Влада Генкина – Миши Кумскова дает неоптимальное решение.

2.3 Алгоритм с дополняющими цепочками

Будем ставить ребят в пары:

Шаг 1. Найдем какое-нибудь разбиение на пары. Просто будем перебирать мальчиков и смотреть, есть ли свободная девочка, с которой он дружит. Таким образом, мы получим несколько пар. Если все мальчики оказались задействованы, то мы нашли оптимальное решение. Если нет, то переходим к пункту 2.

Шаг 2. У нас есть некоторое число незадействованных мальчиков. Попробуем сделать перестановку в парах, которая позволит добавить еще одну пару. Для этого попробуем найти такую цепочку:

(свободный мальчик m_1**)** $\to$
$\to$ **(девочка** d_1**)** $\Rightarrow$ **(её мальчик** m_2**)** $\to$
$\to$ **(свободная девочка** d_2**)**

или более длинную цепочку:

(свободный мальчик m_1**)** $\to$
$\to$ **(девочка** d_1**)** $\Rightarrow$ **(её мальчик** m_2**)** $\to$
$\to$ **(девочка** d_2**)** $\Rightarrow$ **(её мальчик** m_3**)** $\to$
$\to$ **...** $\to$ **(свободная девочка** d_k**)**

В этой цепочке соседние элементы дружат. Двойная стрелка $\Rightarrow$ от девочки к мальчику означает, что на текущий момент эта пара у нас выбрана, а одиночная стрелочка означает, что соответствующая пара дружит, но не выбрана.

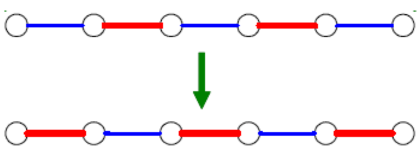
Если мы найдем цепочку, которая начинается со свободного мальчика и заканчивается свободной девочкой, то мы сможем сделать следующую перестановку в парах: жирные стрелочки поменять на нежирные, а нежирные сделать жирными. Например, преобразовать цепочку

$m_1 \to d_1 \Rightarrow m_2 \to d_2 \Rightarrow m_3 \to d_3$

в цепочку

$m_1 \Rightarrow d_1 \to m_2 \Rightarrow d_2 \to m_3 \Rightarrow d_3$ **.**

Заметьте, что число двойных стрелок (число выбранных пар)стало больше на одну – было две, а стало три.



Пока такие цепочки есть, мы можем увеличивать число пар. Но в какой-то момент мы обнаружим, что таких цепочек больше нет, и мы не можем увеличить число пар.

Оказывается, такой способ позволяет получить максимально возможное количество пар. Мы докажем это позже.

Задача Проследите работу алгоритма с дополняющими цепочками для графов, где неправильно работают жадный алгоритм и алгоритм Влада Генкина – Миши Кумскова.

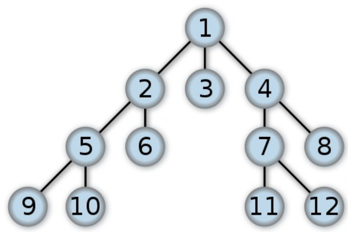
2.4 Как находить цепочки?

Составим граф, в котором из мальчиков исходят ребра (линии), соответствующие дружбе, а из занятых девочек — только одно ребро (линия), ведущее к мальчику, который с ней в паре на текущий момент.

Нам нужно найти цепочку от одного из свободных мальчиков к одной из свободных девочек.

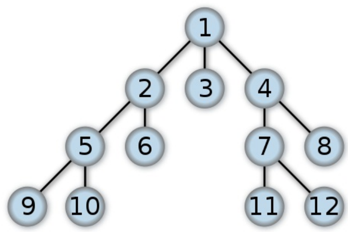
Для этого имеет смысл использовать поиск в ширину в графе. Его также называют "волновым алгоритмом". В начале работы алгоритма "фронт волны"состоит из свободных мальчиков. На первом шаге мы находим всех девочек, которые дружат с одним из свободных мальчиков. Если среди этих девочек есть свободные, то нужная цепочка найдена. На следующем шаге мы берём мальчиков, которые находятся в парах (на текущий момент) с найденными девочками. Эти мальчики представляют собой следующий фронт волны. Тех мальчиков и девочек, через которых прошёл фронт волны, нужно помечать, чтобы второй раз их не рассматривать. В конце концов, мы либо доберемся до свободной девочки, либо на некотором шаге не найдем ничего нового, и фронт волны окажется пустым.

На рисунке показано, в каком порядке обход в ширину просматривает вершины дерева. Мы можем считать, что вершины 1, 5, 6, 7, 8 – мальчики, остальные – девочки.



Заметим, что при реализации этого алгоритма удобно использовать структуру данных "очередь". Встретив при обходе нового ребёнка, будем помещать его в конец этой очереди. А из начала очереди будем последовательно доставать ребенка, чтобы проверить, куда можно сделать следующий шаг в цепочке. И так далее: новеньких найденных (тех, которые еще не были в очереди) будем помещать в конец очереди, не забыв при этом пометить их как "положенные в очередь".

Упражнение Как узнать, соединены ли две вершины графа путём? Опишите, как алгоритм обхода в ширину работает для графа, изображённого на рисунке: в каком порядке обходятся вершины? в каком порядке они добавляются и удаляются из очереди на обход?



Задача Предложите способ, как найти самый “толстый” путь по трубам от истока к стоку? Толщина пути равна минимуму из пропускных способностей труб, по которым он идет.
Подсказка: свести к многократному обходу графа.

Замечание В графе дружб, изображающем нашу задачу, из каждой занятой девочки исходит только одно ребро – к её текущему мальчике-паре, поэтому вместо девочки в очередь можно сразу помещать её пару – мальчика.

У нас остался неясным важный вопрос: всегда ли последовательная оптимизация цепочек приведёт к максимально возможному количеству пар?

3 Отступление: опять доминошки

Пусть чёрные клетки будут мальчиками, а белые – девочками, а соседство клеток обозначает дружбу, тогда задача про доминошки превращается в задачу про максимальное паросочетание. Поиск указанной выше цепочки из мальчиков и девочек соответствует последовательности соседних клеток фигуры. В этой последовательности первая и последняя клетка свободны, все внутренние клетки покрыты доминошками, причём каждый чётный шаг в этой цепочке делается вдоль доминошки. На рисунке показано, как работает шаг оптимизации цепочки, приводящий к увеличению числа доминошек.

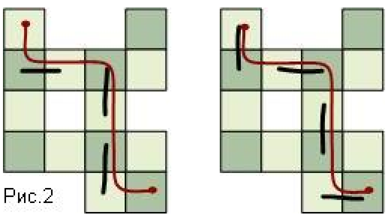


Рис.2

4 Задача про максимальный поток в трубопроводе

Пусть у нас есть система труб, соединённых друг с другом. Её можно представлять как множество узлов, соединённых линиями. Линии – трубы, узлы – это места соединения труб.

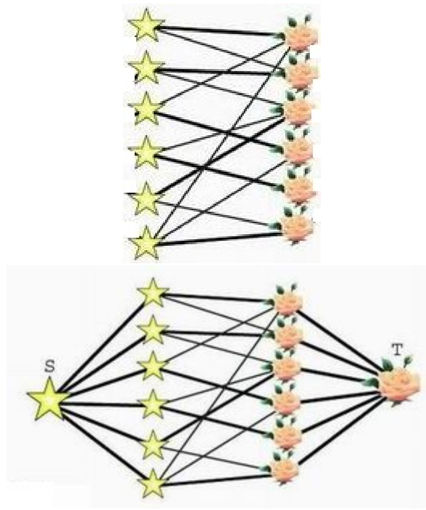
Для каждой линии указана её пропускная способность, то есть сколько единиц жидкости эта труба может пропускать в единицу времени. Среди узлов один помечен как исток, а другой как сток.

Мы собираемся решить задачу: найти максимальное количество единиц жидкости, которую можно пропускать через эту систему труб от истока к стоку.

5 О дружбе и трубах

Покажем, что задача про максимальное паросочетание является частным случаем задачи про максимальный поток. Возьмём двудольный граф, вершины которого есть мальчики и девочки, а рёбра обозначают дружбу. Глядя на этот граф, построим сеть трубопроводов. Изобразим каждую дружбу трубой от мальчика к девочке; пропускная способность такой трубы равна 1.

Дополним двудольный граф дружб двумя вспомогательными вершинами: истоком S и стоком T. Соединим каждого мальчика трубой с истоком S, а девочек со стоком T. Пропускная способность этих труб также равна 1.



В вершину, изображающую мальчика, может втекать не больше одной единицы жидкости, так как пропускная способность подводящей трубы равна 1. Соответственно, выходящий поток из такой вершины не больше 1. Для девочек аналогично: выходящий поток ограничен пропускной способностью труб, соединяющих девочек со стоком: 1; значит и входящий поток не более 1.

Включим подачу воды в истоке S. Потоки в трубах считаем целыми. Поток по трубе между мальчиком и девочкой означает, что мы выбрали их как пару. Целочисленность потока означает, что мы не можем половину мальчика поставить в пару с одной девочкой, а другую половину – с другой.

Получается, что в такой системе трубопроводов любой целочисленный поток изображет пароосочетание. И наоборот: любое паросочетание можно представить потоком. Тогда максимальный поток от S к T в точности равен максимальному числу пар.

6 Решение о максимальном потоке

Решим задачу о нахождении максимального потока в сети трубопроводов, имеющей один исток и один сток. Пропускные способности труб произвольны. Тогда мы автоматически научимся решать и частные случаи этой задачи: поиск максимального паросочетания и поиск максимального покрытия доминошками.

Упражнение Покажите, что для поиска максимального потока, сеть труб с несколькими истоками и несколькими стоками можно описать с помощью подходящей сети с одним истоком и одним стоком.

6.1 Увеличение потока

Метод Форда-Фалкерсона. Пусть у нас есть некоторый поток, который течёт от истока к стоку. Попробуем увеличить его. Для этого найдём путь от истока к стоку по трубам, которые еще не до конца заполнены жидкостью, то есть пропускные способности кусочков труб в этом пути больше, чем поток жидкости, который по ним течёт. По найденному пути мы пустим дополнительный поток жидкости, максимально возможный. В результате этого одна из труб заполнится жидкостью до конца и через неё в следующий раз мы уже не сможем пропустить жидкость в этом направлении. Зато в обратном направлении её пропускная способность как бы удвоилась. Удобно ввести обобщённые пропускные способности труб.

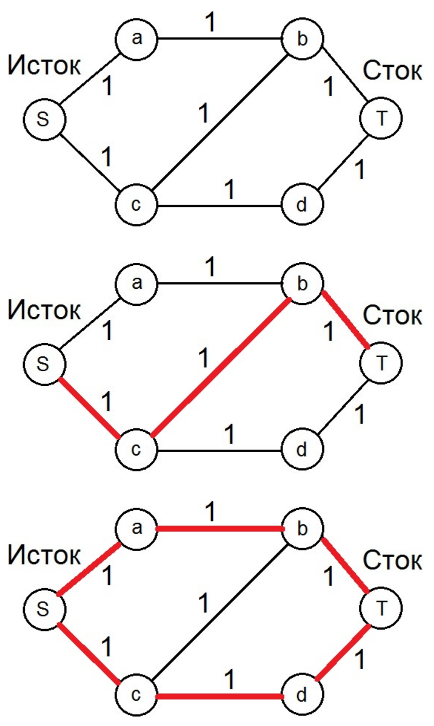
Обобщённая пропускная способность трубы – это два числа, которые показывают проводимость в прямом и обратном направлении. Если труба может проводить не более 2 единиц жидкости в единицу времени, то мы будем писать, что её обобщенная пропускная способность равна [-2,2]. У каждой трубы одно направление условно обозначим как положительное, а другое – как отрицательное. Если в отрицательном направлении труба проводит 5 единиц, а в положительном – 10 единиц жидкости, то её пропускная способность равна [-5,10].

Пусть по трубе [-10,10] пустили поток жидкости величины 5 в положительном направлении. Это можно отобразить как изменение её обобщенной проводимости: теперь в положительном направлении можно пустить еще максимум 5 единиц, зато пропускная способность в другую сторону увеличилась. Она стала равна 15: мы можем убрать положительный поток в 5 единиц и еще дополнительно пустить 10 единиц в отрицательном направлении. Итого, можно считать, что труба [-10,10] с потоком 5 в положительном направлении, превратилась в трубу [-15,5].

Таким образом, при построении алгоритма мы можем работать с обобщёнными пропускными способностями. Каждый раз, когда мы пускаем жидкость по найденному пути, мы будем просто менять обобщённые пропускные способности труб и получать новую сеть труб (с новыми параметрами), по которым (как будто бы) ничего не течёт.

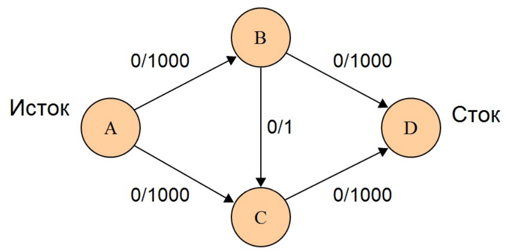
Итак, мы ищем путь от истока к стоку, двигаясь по трубам в направлении, в котором они могут пропустить какое-то количество жидкости (их текущая пропускная способность в этом направлении положительна). Пускаем по этому пути максимально возможный поток жидкости, после чего одна из труб в этом пути оказывается заполненной до конца (самое узкое звено). Пересчитываем обобщенные пропускные способности и снова повторяем шаг поиска пути. Так делаем, пока такой путь существует. В какой-то момент мы не сможем добраться по пропускающим трубам от истока к стоку.

Задача Проследите изменение обобщенных пропускных способностей при увеличении потока в сети на рисунке. Числа показывают обычные пропускные способности рёбер. Жирные линии красные линии — трубы, по которым пропущен единичный поток.

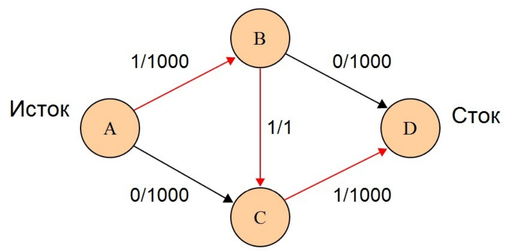


Задача: Разбиение на пары и поток в трубах Проведите рассуждения с увеличением потока для сети труб, описывающей граф дружб, дополненный истоком и стоком. Пропускные способности всех рёбер равны 1.
- Какой обобщенный поток в трубе соответствует невыбранной паре (мальчик – девочка)?
- Какой поток соответствует выбранной паре?
- Как при увеличении потока будут меняться обобщённые пропускные способности рёбер? Разберите случаи:
- Можно просто добавлять новые пары (пропускать дополнительный поток по свободным трубам);
- Для увеличения числа пар надо разбить какие-то существующие пары. Что это означает для потока?

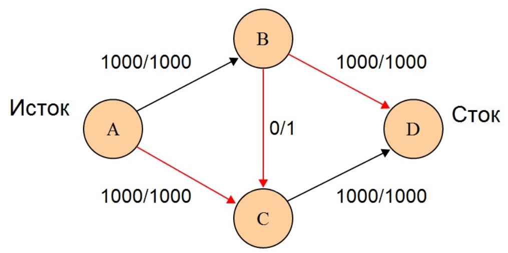
Задача Проследите изменение обощенных пропускных способностей при увеличении потока в сети, изображенной на рисунке. Числа $1/1000$ означают поток величины 1 по ребру с пропускной способностью 1000.



Сколько шагов может потребоваться алгоритму для нахождения максимального потока:
а) предложите удачный выбор увеличения потока для данного графа; сколько шагов потребуется?
б) при самом неудачном способе (изображен на рисунке).



много шагов увеличения потока...



Как мы видели из последней задачи, необходимое количество шагов увеличения потока сильно зависит от того, каким образом мы выбираем путь для увеличения потока.

Один из вариантов – искать методом поиска в ширину путь от истока к стоку, по которому возможно пропустить дополнительный поток. В случае целых весов это гарантирует, что алгоритм сделает не слишком большое число шагов увеличения потока. Количество зависит от подробностей реализации.

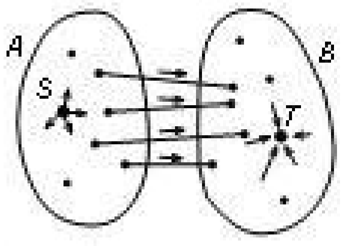
Для сведения. Для сети из n вершин и m рёбер, при выборе дополняющего потока с помощью поиска в ширину, достаточно $\sim n * m^2$ шагов.
(Поиск в ширину требует $\sim m$ операций.)
Лучшие из придуманных алгоритмов работают гораздо быстрее.

6.2 Доказательство оптимальности

Почему метод Форда-Фалкерсона найдёт максимально возможный поток?

Итак, мы ищем путь от истока к стоку по трубам, двигаясь по трубам в том направлении, в котором они имеют положительную проводимость. На последнем шаге алгоритма мы обнаруживаем, что такого пути нет.

Рассмотрим множество вершин A, до которых мы можем добраться от истока. Обозначим множество остальных вершин как B. Исток находится в множестве A, а сток – в множестве B.



Рассмотрим все трубы, которые начинаются в одной из вершин A, а заканчиваются в одной из вершин B.

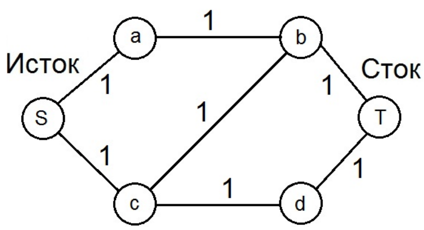
Через каждую из этих труб течёт максимально возможный поток, причём жидкость течёт в направлении от A к B. Иначе мы могли бы попасть из некоторой вершины из A в некоторую вершину из B по не заполненной до конца трубе.

Суммарный поток, идущий по этим трубам, в точности равен их суммарной проводимости и текущему потоку из истока к стоку.

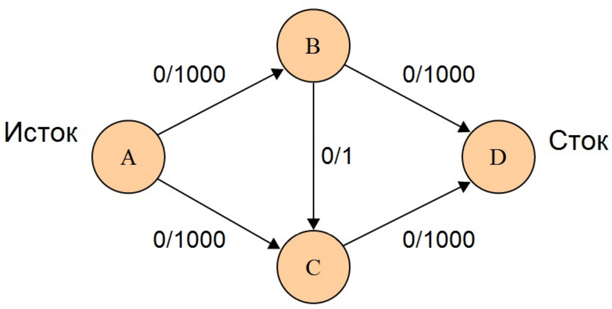
Вся жидкость, которая появляется в истоке S, растекается по вершинам множества A, затем без потерь перетекает по указанным трубам в множество B и в конечном итоге попадает в сток T.

Задача На заданной сети найдите максимальный поток и разрез с минимальной пропускной способностью.

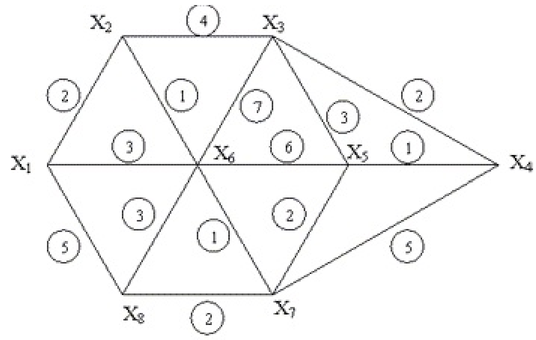
а)



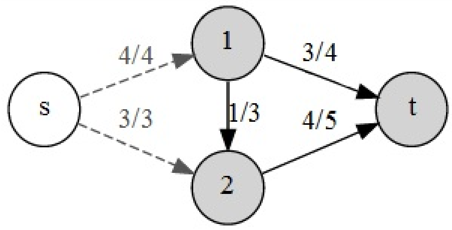
б)



в) Найти максимальный поток из X1 в X4. Найдите разрез с минимальной пропускной способностью.



г) Проверьте равенство максимального потока и разреза с минимальной пропускной способностью:



Задача Рассмотрим сеть с несколькими источниками и стоками, каждый производит или потребляет свое количество единиц потока. Предложите способ узнать, возможно ли в такой сети весь поток из источников передать в сток? Что нужно изменить в алгоритме Форда-Фалкерсона, чтобы получить решение?

7 А теперь – своими руками

На сайте EI JUDGE http://acm.mipt.ru/judge (постоянно действующее соревнование по программированию среди школьников и студентов) есть описанная нами задача про замощение клеточной фигуры доминошками (задача 068:

`http://acm.mipt.ru/judge/bin/problems.pl?problem=068`

Задача Попробуйте написать программу, которая решает задачу про доминошки, и послать код на проверку. Можете найти и взять готовую реализацию алгоритма Форда-Фалкерсона.

8 Задача про коробки

Вот ещё одна интересная задача, для решения которой применяется поиск максимального паросочетания.

Задача У Пети накопилось много пустых картонных коробок. Они ему ещё сослужат службу в будущем, но сейчас он хотел бы сдать их в камеру хранения. Упакуйте коробки друг в друга, получив минимальное количество коробок. Никакие две коробки не поместятся ни в какую другую, если их поставить рядом. Коробка со сторонами x_1, x_2, x_3, $x_1 \geq x_2 \geq x_3$, помещается в коробку со сторонами y_1, y_2, y_3, (считаем $y_1 \geq y_2 \geq y_3$), если $x_1 < y_1$, $x_2 < y_2$, $x_3 < y_3$.

Входные данные: число коробок и их размеры.

Хотим получить минимальное число коробок, в которые все остальные можно поместить, и способ укладки.

Обсуждение, подсказки Что значит, что никакие две коробки не влезают рядом друг с другом в третью коробку? Придумайте систему коробок и изобразите её с помощью графа вложений. Что стало бы с этим графом, если несколько коробок можно было положить рядом?

Коробка может выступать в двух качествах? Попробуйте, глядя на граф вложений коробок, нарисовать двудольный

граф. Чему будут соответствовать паросочетания в этом графе? Максимальные паросочетания?
Посмотрите обсуждение задачи на форуме. Найдите ошибочные и правильные в рассуждении участников форума.

`http://forum2004.algolist.ru/viewtopic.php?t=1706`

Напишите программу и попробуйте послать код на проверку. Автоматическая проверка:

`http://acm.mipt.ru/judge/problems.pl?problem=013`

9 Что почитать

Основу текста и план изложеня: доминошки – паросочетания – трубы – взял из статьи

Ворожцов Артём Викторович, "О доминошках, дружбе мальчиков и девочек и трубопроводе

`http://potential.org.ru/Info/ArtDt200503291423PH3C1J3`

Другое изложение метода Форда-Фалкерсона:

`http://kvant.mirror1.mccme.ru/1970/04/parosochetaniya_i_transportnye.htm`