Эйлеровы обходы.

Определение. Замкнутый маршрут P в графе G называется *эйлеровым обходом*, если каждое ребро графа встречается в P ровно один раз.

Определение. Назовем граф без изолированных вершин *эйлеровым*, если в нем есть эйлеров обход.

Утверждение. Эйлеров граф связен.

Теорема. Для связного графа G эквивалентны следующие утверждения.
I G – эйлеров.
II Каждая вершина в G имеет четную степень.
III Множество ребер графа G можно разбить на циклы.

Доказательство. I ⇒ II. При каждом прохождении эйлерова маршрута через любую вершину используется 2 ребра.

II ⇒ III. Эта импликация верна и для несвязного графа. Пусть это не так и G — минимальный, по количеству рёбер, контрпример. Тогда $|E(G)| > 0$. Отбросим все изолированные вершины. Получим граф с минимальной степенью не меньше 2. Следовательно, в нем есть цикл C. Рассмотрим граф G_1, полученный удалением всех рёбер цикла. Но $|E(G_1)| < |E(G)|$ и по предположению G_1 можно разбить на циклы. Противоречие.

III ⇒ I. Индукция по количеству циклов p. При $p = 1$ этот цикл и будет искомым эйлеровым маршрутом.

Пусть утверждение верно при любом количестве циклов меньшем p. Пусть граф G разбит на p циклов C_1, C_2,…, C_p. Рассмотрим граф G_1, полученный удалением всех рёбер цикла C_1. Тогда каждая компонента связности H графа G_1 разбивается меньше чем на p циклов и по предположению индукции имеет эйлеров обход R_H.

Заметим, что каждая компонента H имеет общую вершину с C_1. Построим эйлеров обход исходного графа следующим образом. Двигаясь по циклу C_1 и попав в его очередную вершину v, обходим ранее не пройденную компоненту H, содержащую v (если такая H найдется), следуя R_H, и возвращаемся в v. Затем продолжаем движение по C_1.

Замечание. Поскольку доказательство конструктивное, то его можно реализовать в виде алгоритма построения эйлерова цикла.

Следствие. В связном графе существует маршрут, соединяющий вершины s и t и проходящий по каждому ребру ровно один раз, если и только если степени s и t нечетны, а степени всех остальных вершин — четны.

Гамильтоновы циклы.

Определение. Замкнутый маршрут P в графе G называется *гамильтоновым циклом*, если каждая вершина графа встречается в P ровно один раз.

Определение. Назовем граф *гамильтоновым*, если в нем есть гамильтонов цикл.

Утверждение. Гамильтонов граф двусязен.

Доказательство. Пусть это не так и существует вершина v, после удаления которой, множество вершин распадется на подмножества V_1 и V_2. Тогда любой путь из V_1 в V_2 должен проходить через v. Но тогда вершина v встретится на любом замкнутом обходе вершин как минимум дважды на пути из V_1 в V_2 и обратно.

Достаточное условие гамильтоновости.

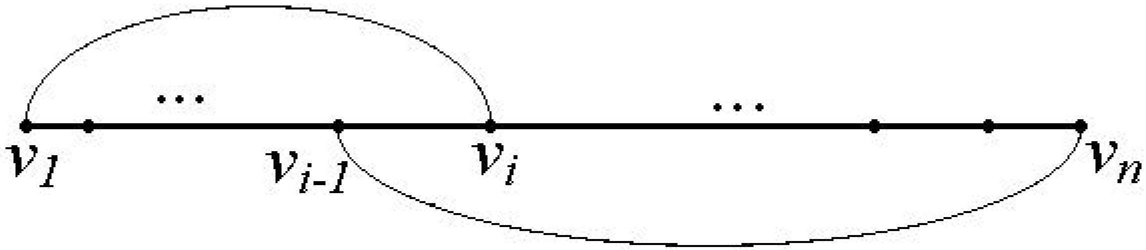
Теорема Дирака. Если граф содержит не менее 3 вершин, и для любой вершины v графа G выполняется неравенство $d(v) \geq |V(G)|/2$, то граф гамильтонов.

Доказательство. Граф связен, иначе каждая компонента связности имеет не менее чем $1+|V(G)|/2$ вершин, что невозможно.

Предположим, что утверждение теоремы неверно и граф G — контрпример. Пусть $P=v_1v_2\ldots v_n$, — самый длинный путь в графе.

Построим гамильтонов путь в подграфе, порожденном вершинами v_1, v_2,… v_n. Так как путь самый длинный, то вершины v_1 и v_n инцидентны только вершинам из P.

Если найдется такое i, $2 \le i \le n$, что v_1 смежна с v_i, а v_n — с v_{i-1}, то цикл $C=v_1 \ldots v_{i-1} v_n \ldots v_i v_1$ — искомый.



Пусть это не так и такой вершины v_i не существует. Тогда v_n не смежна с $d(v_1)$ вершинами пути P и, следовательно, $d(v_n) \le n-1-d(v_1)$. Тогда $|V(G)| \le d(v_1)+d(v_n) \le n-1 \le |V(G)|-1$.

Противоречие. Таким образом, цикл порожденный вершинами $v_1, v_2, \ldots v_n$ существует.

Если $V=\{v_1, v_2, \ldots v_n\}$, то теорема доказана. Пусть это не так. Но граф G связен и тогда существует вершина w, смежная с некоторой вершиной v_i цикла C. Множество рёбер $\{(w,v_i)\} \cup C \setminus \{(v_i, v_{i+1})\}$ образует путь длины $n+1$, противоречие.

Пример. Если на званом обеде каждый знаком не менее, чем с половиной присутствующих, то всем можно сесть за круглым столом так, что по обе стороны от каждого будут сидеть его знакомые.

Замечание. Условие теоремы нельзя ослабить. В двудольном графе $K_{n,n+1}$ гамильтонов цикл должен иметь нечетную длину, что невозможно. Тем не менее, минимальная степень вершины равна $n = \left(\left|V\left(K_{n,n+1}\right)\right| - 1\right)/2$.

Задача. Найти максимальный по количеству рёбер n-вершинный граф, не содержащий клики размера k.

Определение. Обозначим $T_{n,k}$ граф на n вершинах, все вершины которого разбиты на подмножества V_1, V_2,… $V_{k\text{-}1}$ так, что мощность любого подмножества равна $\left\lfloor \frac{n}{k-1} \right\rfloor$ или $\left\lceil \frac{n}{k-1} \right\rceil$, и две вершины соединены ребром тогда и только тогда, когда лежат в разных множествах.

Теорема Турана. Граф $T_{n,k}$ — максимальный по количеству рёбер n-вершинный граф, не содержащий клики размера k.

Замечание. Пусть $n=q(k-1)+r$, где $r<k-1$. Тогда $\left|E\left(T_{n,k}\right)\right|=\frac{(k-2)\left(n^2-r^2\right)}{2k-2}+\frac{r(r-1)}{2}$.

Доказательство. Удалим по одной вершине из подмножеств размерности $\left\lceil\frac{n}{k-1}\right\rceil$. Тогда каждая из оставшихся n-r вершин имеет n-r-q соседей, что в совокупности даст $\frac{(n-r)(n-r-q)}{2}$ рёбер. Каждая удаленная вершина соединена с n-r-q из оставшихся вершин и со всеми удаленными. Это добавляет еще $r(n-r-q)+\frac{r(r-1)}{2}$. Отсюда

$$\left|E\left(T_{n,k}\right)\right|=\frac{(n-r)(n-r-q)}{2}+r(n-r-q)+\frac{r(r-1)}{2}=\frac{(n+r)(n-r-q)}{2}+\frac{r(r-1)}{2}=$$

$$=\frac{(n+r)(k-2)q}{2}+\frac{r(r-1)}{2}=\frac{(k-2)(n+r)q(k-1)}{2(k-1)}+\frac{r(r-1)}{2}=\frac{(k-2)\left(n^2-r^2\right)}{2k-2}+\frac{r(r-1)}{2}.$$

Замечание. При $k\leq 8$ формула для максимального количества рёбер принимает вид $\left|E\left(T_{n,k}\right)\right|=\left\lfloor\frac{(k-2)n^2}{2k-2}\right\rfloor$.

Доказательство Теоремы Турана при k=3.

Пусть граф G искомый, а вершина z имеет в нем максимальную степень d. Обозначим $N(z)$ множество инцидентных z вершин и $Z=V(G)\backslash N(z)$. Очевидно, что нет рёбер соединяющих вершины из $N(z)$, иначе существует цикл длины 3, т.е. клика.

Покажем, что в $Z\backslash\{z\}$ нет внутренних рёбер. Пусть это не так и найдутся вершины x и y, лежащие в $Z\backslash\{z\}$ и соединённые ребром. Тогда общее количество рёбер инцидентных вершинам x и y меньше $2d$. Заменив все рёбра инцидентные вершинам x и y на рёбра множества $\{x, y\}\times N(z)$, получим граф, не содержащий 3-клики, большей размерности.

Следовательно, в Z нет внутренних рёбер и $G=K_{d,n-d}$. Количество рёбер в полном двудольном графе максимально, когда размеры долей равны.

Пример. В круговом турнире участвует 12 команд. Какое минимальное число матчей надо провести, чтобы в подтурнире любых трех команд оказался сыгранным хотя бы один матч? Ответ: $C_{12}^{2}-\left\lfloor\frac{12^{2}}{4}\right\rfloor=30$.

Алгоритмы на графах

Поиск по графу

Задан граф $G = (V, E)$, $A(v), v \in V$ — списки смежности

Найти компоненты связности.

Алгоритм ПОИСК(v)

1. $Q := \{v\}$, пометить v.
2. До тех пор пока $Q \neq \varnothing$ выполнять

Пусть $x \in Q$; удалить x из Q;

Для всех $y \in A(x)$:
если у не помечена, то добавить y в Q и пометить y

Утверждение. Алгоритм ПОИСК помечает все вершины графа, достижимые из v, за $O(|E|)$ элементарных операций.

Доказательство. Пусть V_1 — множество вершин, достижимых из v

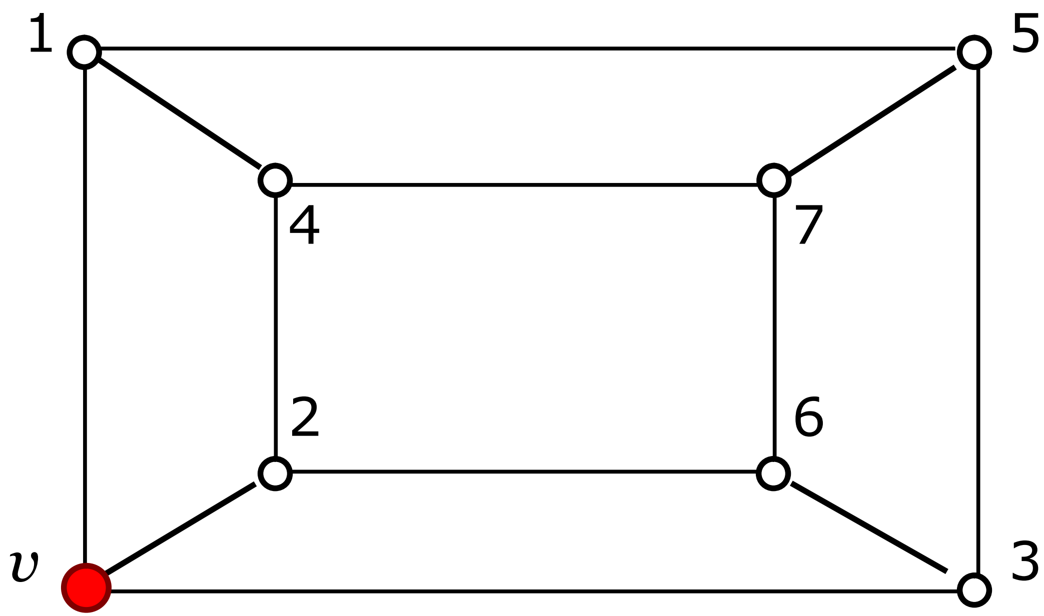
1. Для записи v требуется время $O(1)$.
2. Работа с множеством Q. Добавление и удаление элементов производится $2|V_1|$ раз. Если Q — очередь или стек, то каждое включение или удаление требует $O(1)$ времени.
3. Поиск по спискам смежности. Каждый элемент в списке просматривается не более одного раза. Всего $2|E|$ элементов.

Суммарная оценка $O(|E|)$.

Поиск в ширину

Множество организовано в виде очереди:

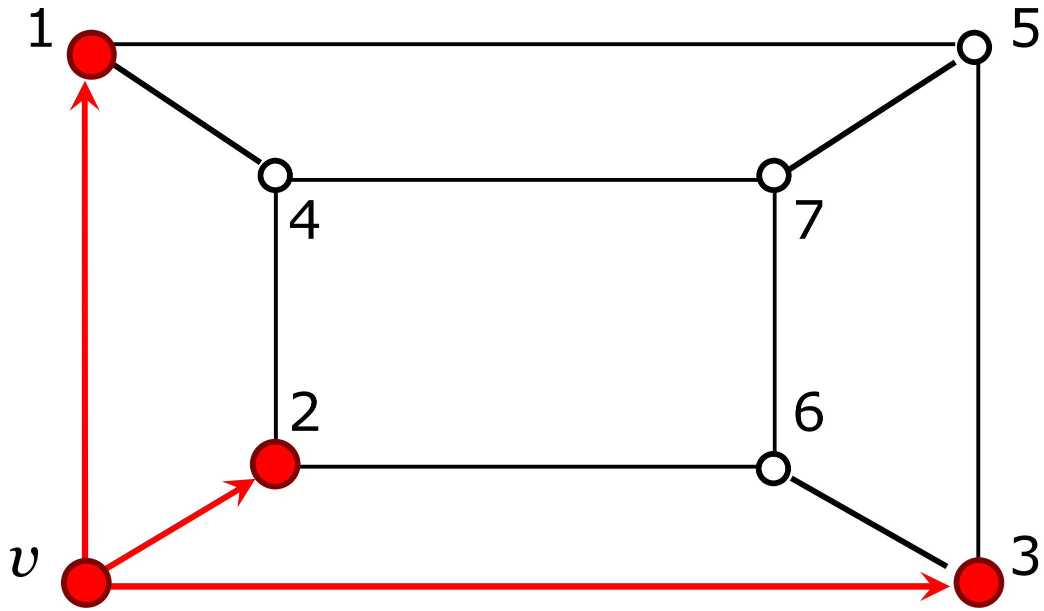
первым пришел — первым ушел.



Поиск в ширину

Множество организовано в виде очереди:

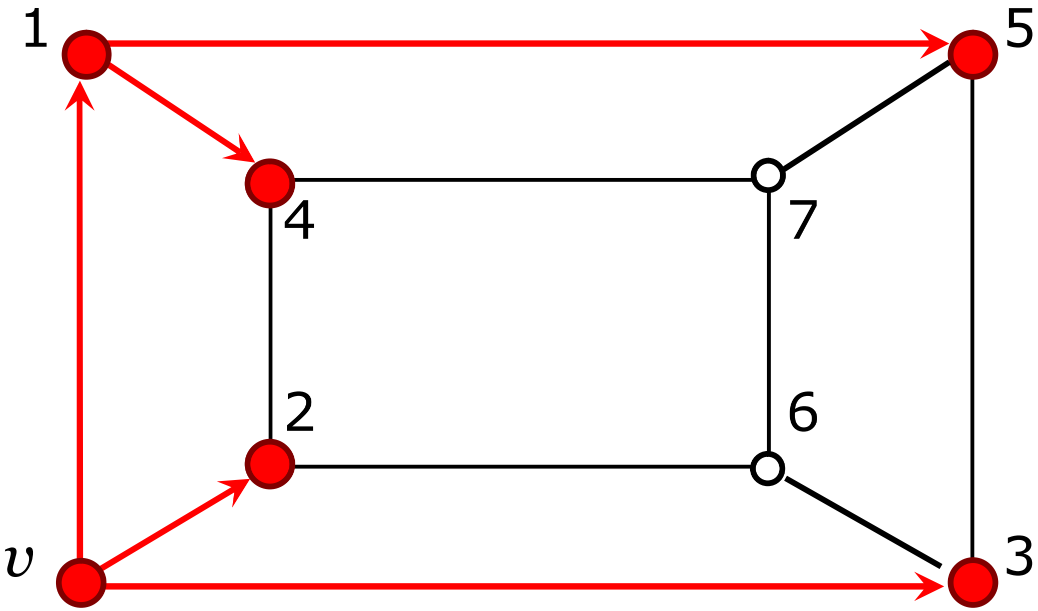
первым пришел — первым ушел.



Поиск в ширину

Множество организовано в виде очереди:

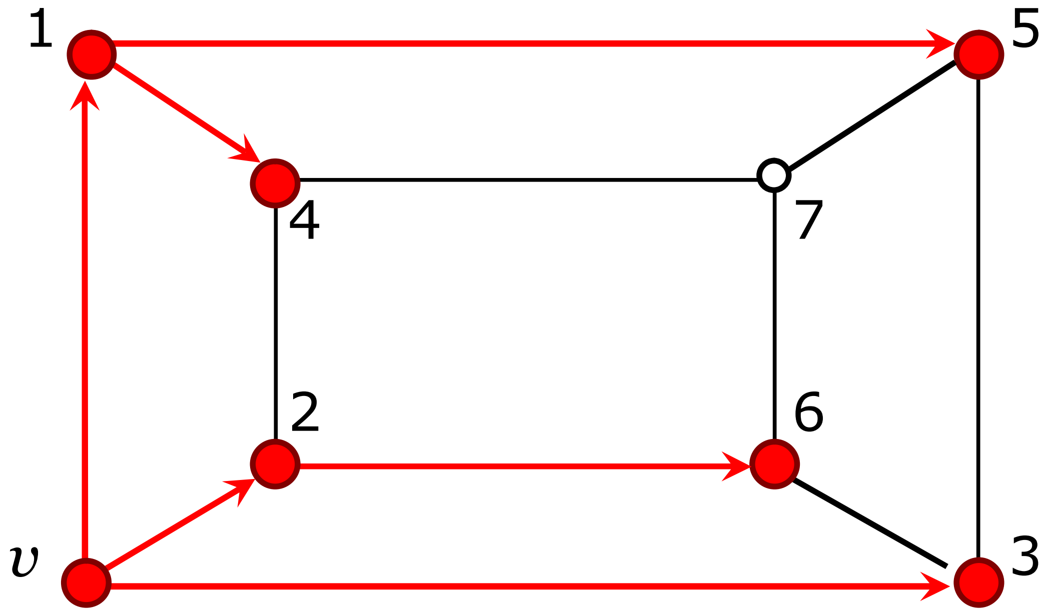
первым пришел — первым ушел.



Поиск в ширину

Множество организовано в виде очереди:

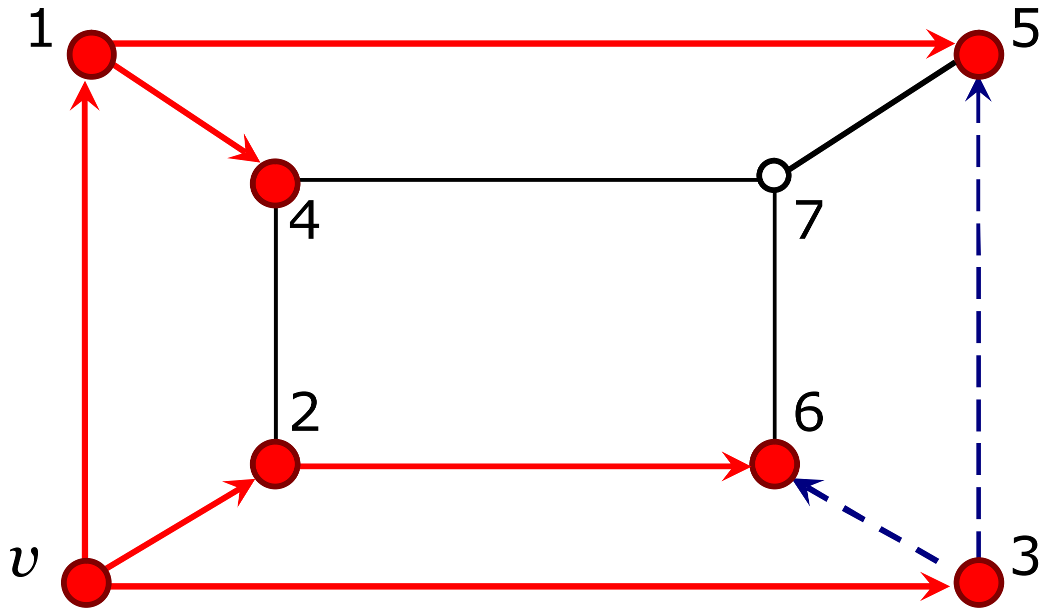
первым пришел — первым ушел.



Поиск в ширину

Множество организовано в виде очереди:

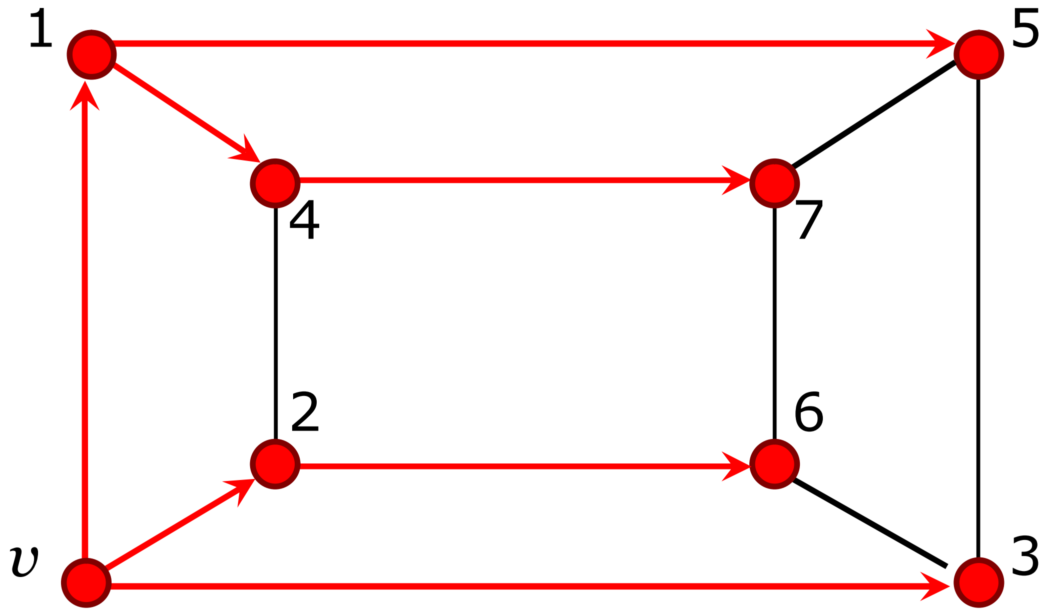
первым пришел — первым ушел.



Поиск в ширину

Множество организовано в виде очереди:

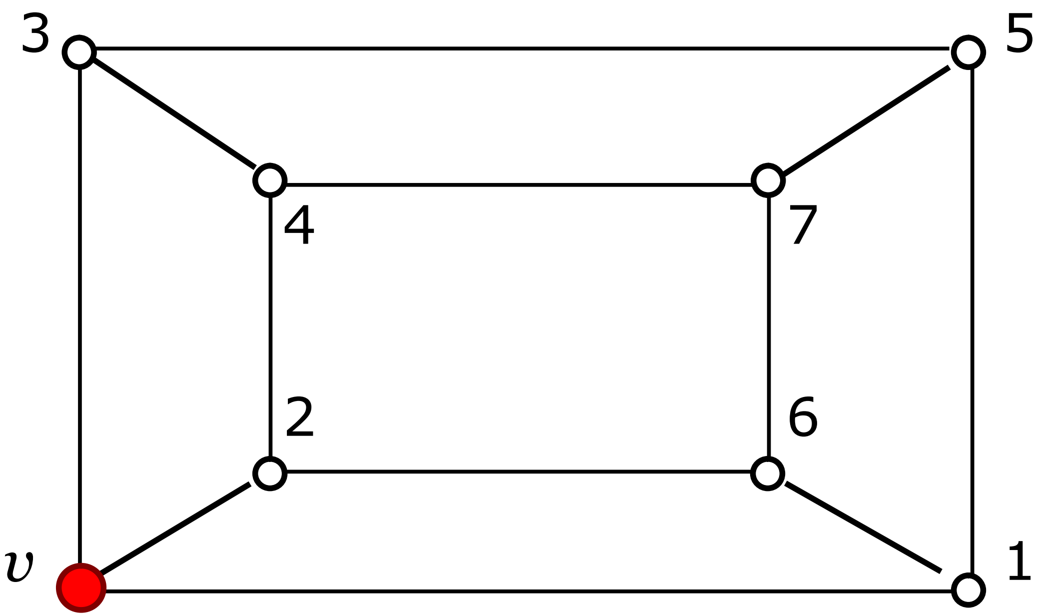
первым пришел — первым ушел.



Поиск в глубину

Множество Q организовано в виде стека:

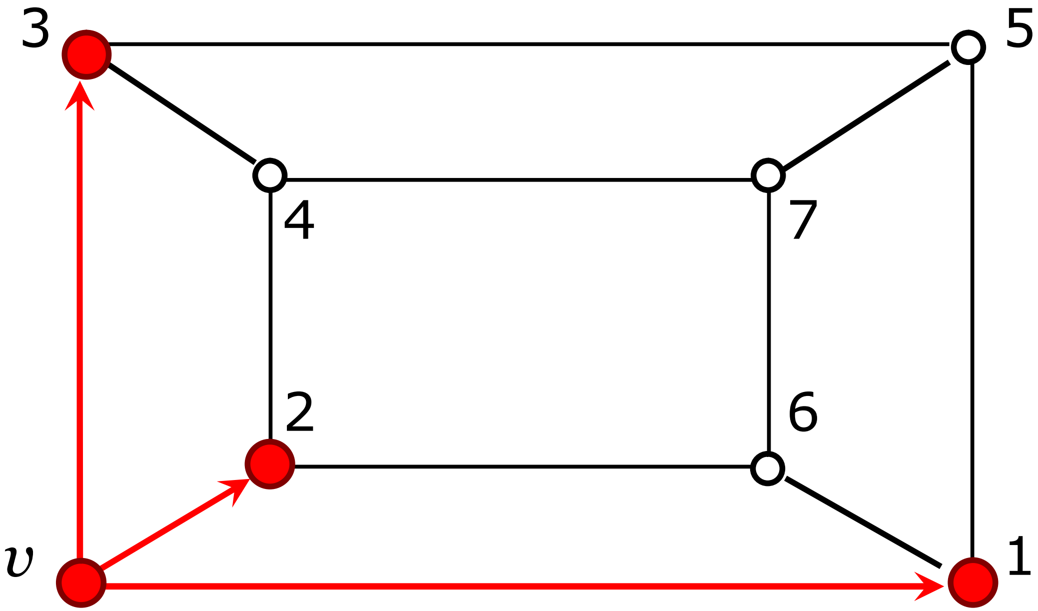
последним пришел — первым ушел.



Поиск в глубину

Множество Q организовано в виде стека:

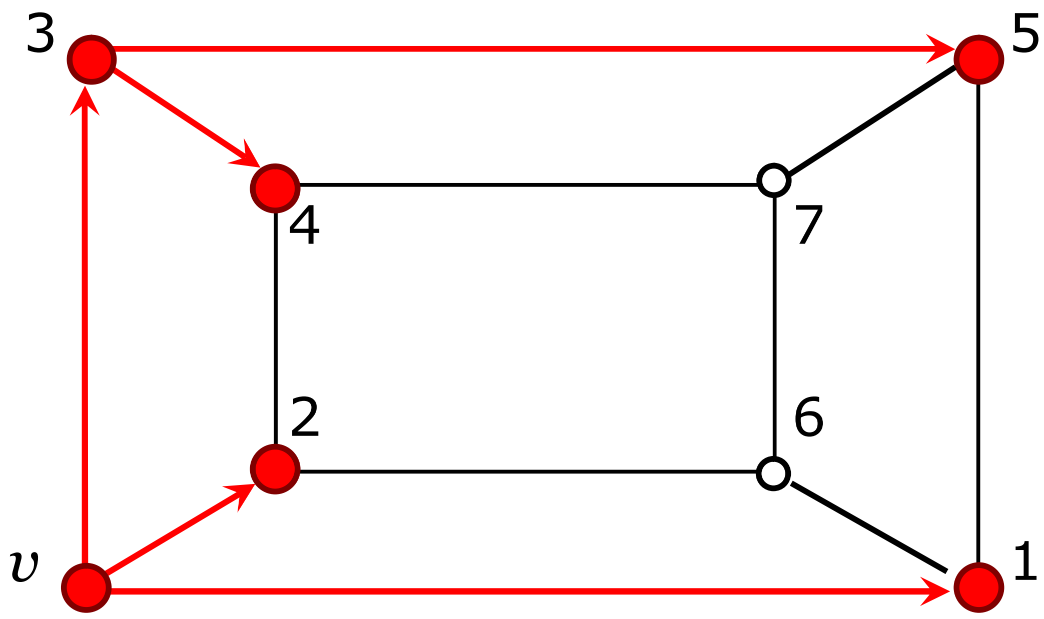
последним пришел — первым ушел.



Поиск в глубину

Множество Q организовано в виде стека:

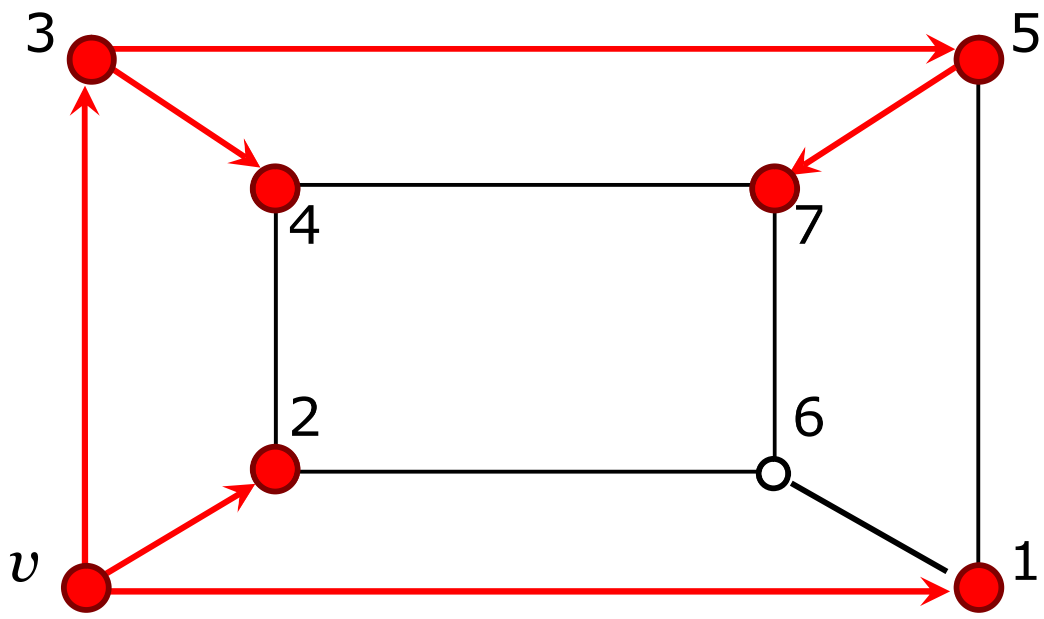
последним пришел — первым ушел.



Поиск в глубину

Множество Q организовано в виде стека:

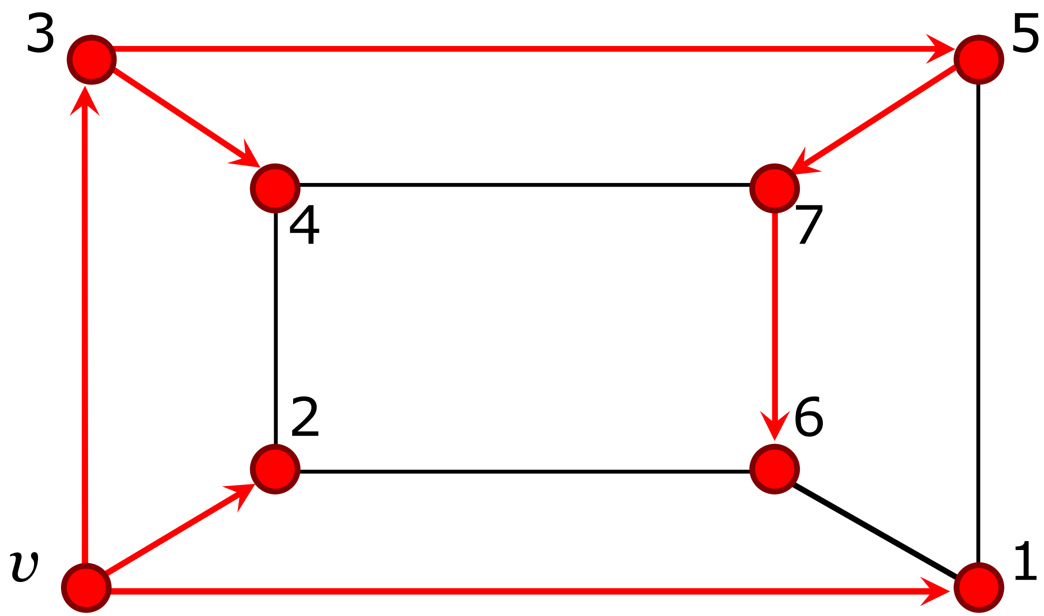
последним пришел — первым ушел.



Поиск в глубину

Множество Q организовано в виде стека:

последним пришел — первым ушел.



Проверка двудольности графа

Задан граф $G = (V, E)$, $A(v), v \in V$ — списки смежности

Задача. Разбить вершины на доли, не содержащие внутренних ребер, или показать, что такого разбиения не существует.

Алгоритм Двудольность

1. Выбрать произвольную непомеченную вершину v.
2. $Q := \{v\}$, пометить $\varphi(v) = 0$.
3. До тех пор пока $Q \neq \varnothing$ выполнять.

Пусть $x \in Q$; удалить x из Q;

Для всех $y \in A(x)$:

если y помечена и $\varphi(y) = \varphi(x)$, то граф не двудолен,

если y не помечена, то добавить y в Q и пометить $\varphi(y) = 1 - \varphi(x)$.

4. Если есть непомеченные вершины вернуться к пункту 1.

Поиск цикла в графе

Задача. Найти цикл или доказать, что цикла не существует.

Алгоритм Цикл

1. Выбрать произвольную непомеченную вершину v.
2. $Q := \{v\}$, пометить $\varphi(v) = 0$.
3. До тех пор пока $Q \neq \varnothing$ выполнять

Пусть $x \in Q$; удалить x из Q;

Для всех $y \in A(x)$:

если y не помечена, то добавить y в Q и пометить $\varphi(y) = x$,

если y помечена то по меткам восстановить пути из x и y до их первого общего предка, добавить к полученному пути ребро (x,y) и выйти из процедуры.

4. Если есть непомеченные вершины, вернутся к пункту 1.

Поиск эйлерова обхода.

Задан граф $G = (V, E)$ без изолированных вершин, $A(v)$, $v \in V$ — списки смежности.

Задача. Построить Эйлеров обход или доказать, что его не существует.

Алгоритм Эйлер (G, M_G)

Шаг 1. Найти цикл M_G в графе G. Если цикла не существует, то exit.

Шаг 2. Найти вершину v из $V(M_G)$ смежную с $V(G) \setminus V(M_G)$. Если такой вершины не существует то exit.

Шаг 3. Найти компоненту связности H_v в графе $G \setminus M_G$, содержащую вершину v. Вызвать процедуру Эйлер(H_V, M_H).

Шаг 4. Заменить в обходе M_G вершину v на обход M_H. Если M_G не содержит все вершины вернуться на шаг 2.

Утверждение. Алгоритм Эйлер (G, M_G) находит эйлеров обход M_G графа G, за $O(|E|^2)$ элементарных операций.

Доказательство. Алгоритм содержит рекурсивное обращение к процедуре. Оценим суммарное количество таких обращений. При выполнении последовательности шагов 2–4 процедура вызывается максимум столько же раз сколько рёбер в найденном на шаге 1 цикле. Суммарное количество ребер в графах, переданных в процедуры, равно количеству рёбер в исходном графе минус количество рёбер цикла, найденного на шаге 1. Следовательно, суммарное количество обращений к процедуре не больше $|E|$.

Шаг 1 требует $O(|E|)$ операций.

Шаг 2 выполняется по одному разу для каждой компоненты связности и требует, в совокупности, $O(|E|)$ операций.

Шаг 3 выполняется по одному разу для каждой компоненты связности и требует, в совокупности, $O(|E|)$ операций, включая рекуррентное обращение к процедуре.

Шаг 2 выполняется по одному разу для каждой компоненты связности и требует, в совокупности, $O(|V|)$ операций.