Алгоритмы на графах

Лекция 3.

Поиск в глубину.

Адигеев Михаил Георгиевич

2023

План лекции

1. Обход в глубину.
2. Применение обхода в глубину для решения задач на графах.
   - ✓ Обнаружение циклов
   - ✓ Определение двудольности графа

Обход в глубину

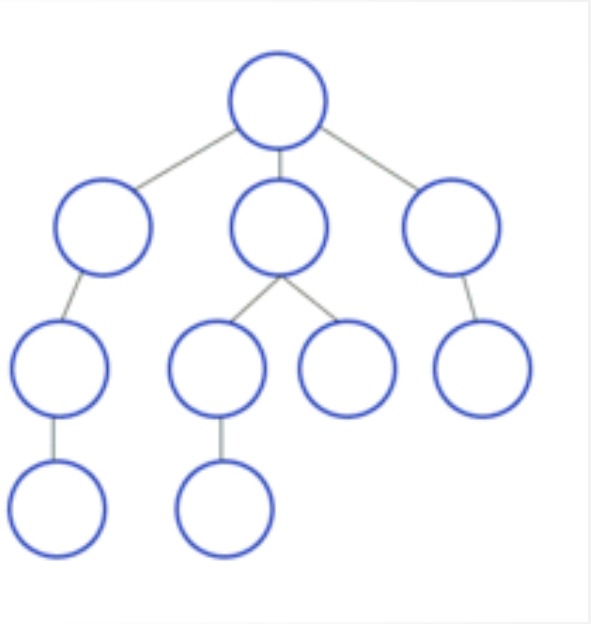
Общие принципы обхода

- Мы должны *обойти* все вершины графа в *определённом* порядке и *обработать* их.
- В процессе обхода для каждой вершины мы в какой-то момент посещаем её в первый раз, потом можем несколько раз возвращаться в неё, и в какой-то момент (обычно при первом или при последнем посещении) мы её обрабатываем.
- Для того чтобы корректно обработать вершину (а также инцидентные ей рёбра и смежные вершины), мы должны отслеживать *статус* вершины:
  1) Непосещённая
  2) Посещённая
  3) Обработанная.

Обход в глубину

Принцип обхода в глубину:

посетив вершину v, последовательно выполняем *обход в глубину* для каждого ещё не посещённого соседа.



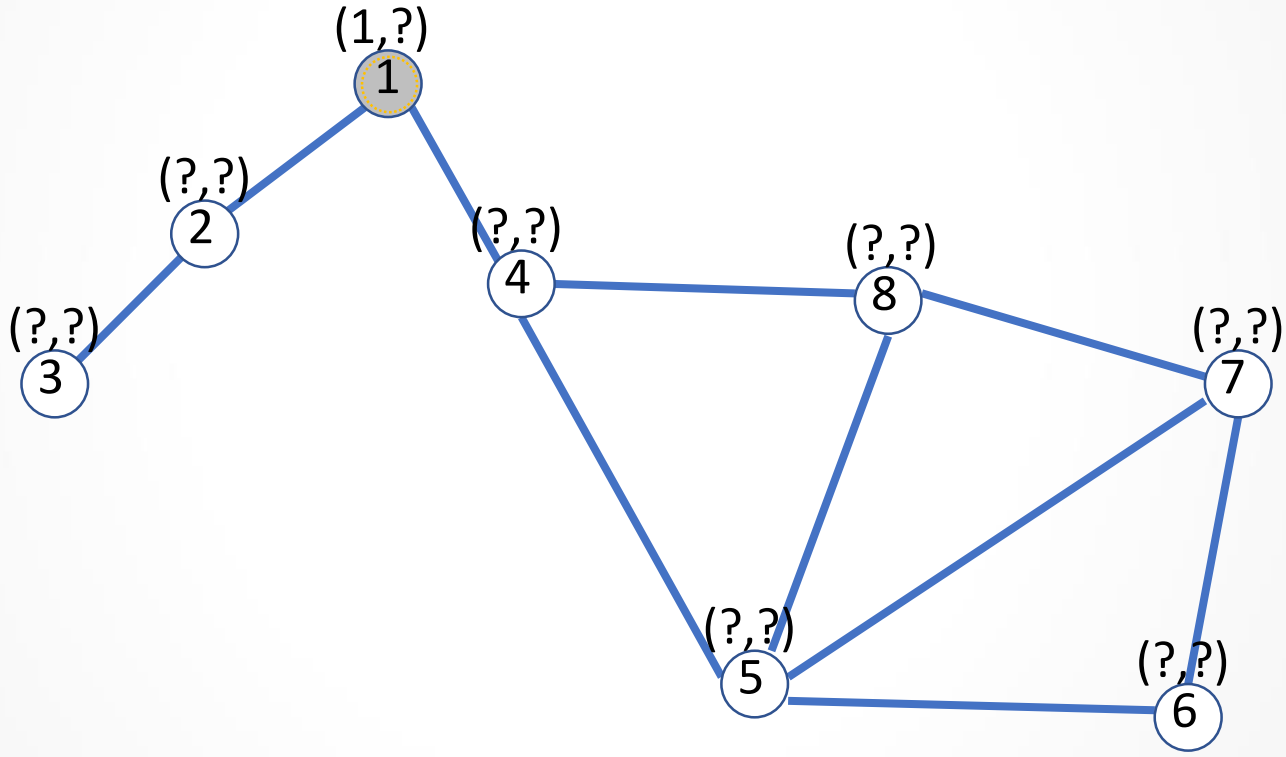
https://en.wikipedia.org/wiki/Depth-first_search

Для того чтобы запоминать порядок посещения вершин нам понадобится структура данных «Стек» (stack) – явный или неявный.

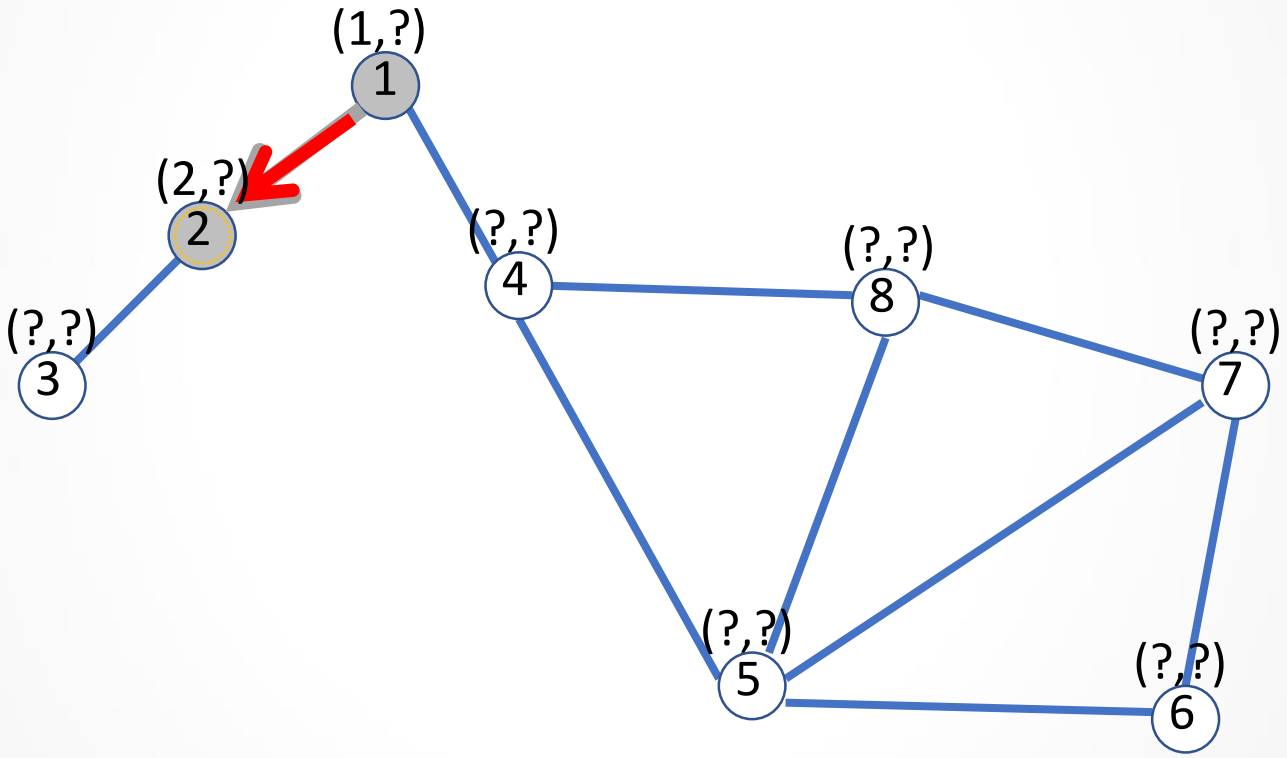
Обход в глубину

- При обходе в глубину будем для каждой вершины фиксировать время входа (`in`) и время выхода (`out`).
- ‘Время’ – это номер шага при выполнении обхода (сквозной счётчик).
- Эти показатели понадобятся для решения некоторых задач на графах с помощью обхода в глубину.

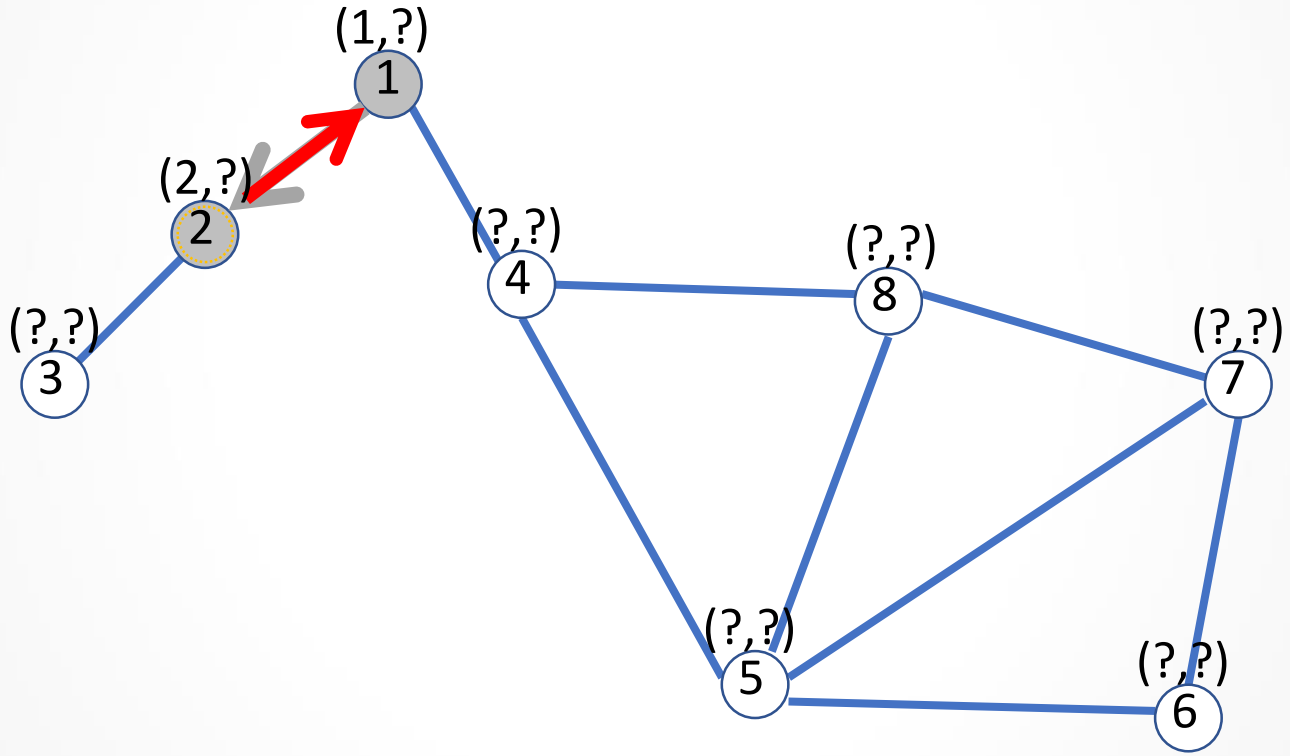
Обход в глубину: пример



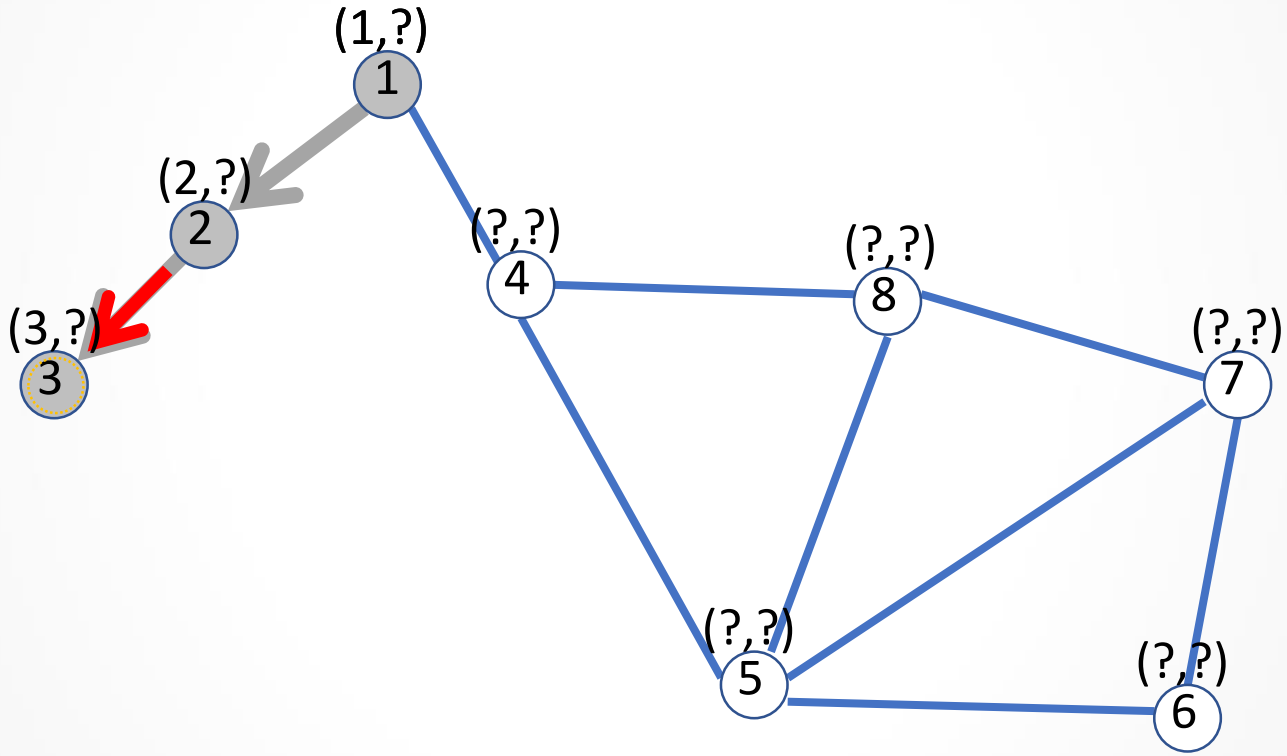
Обход в глубину: пример



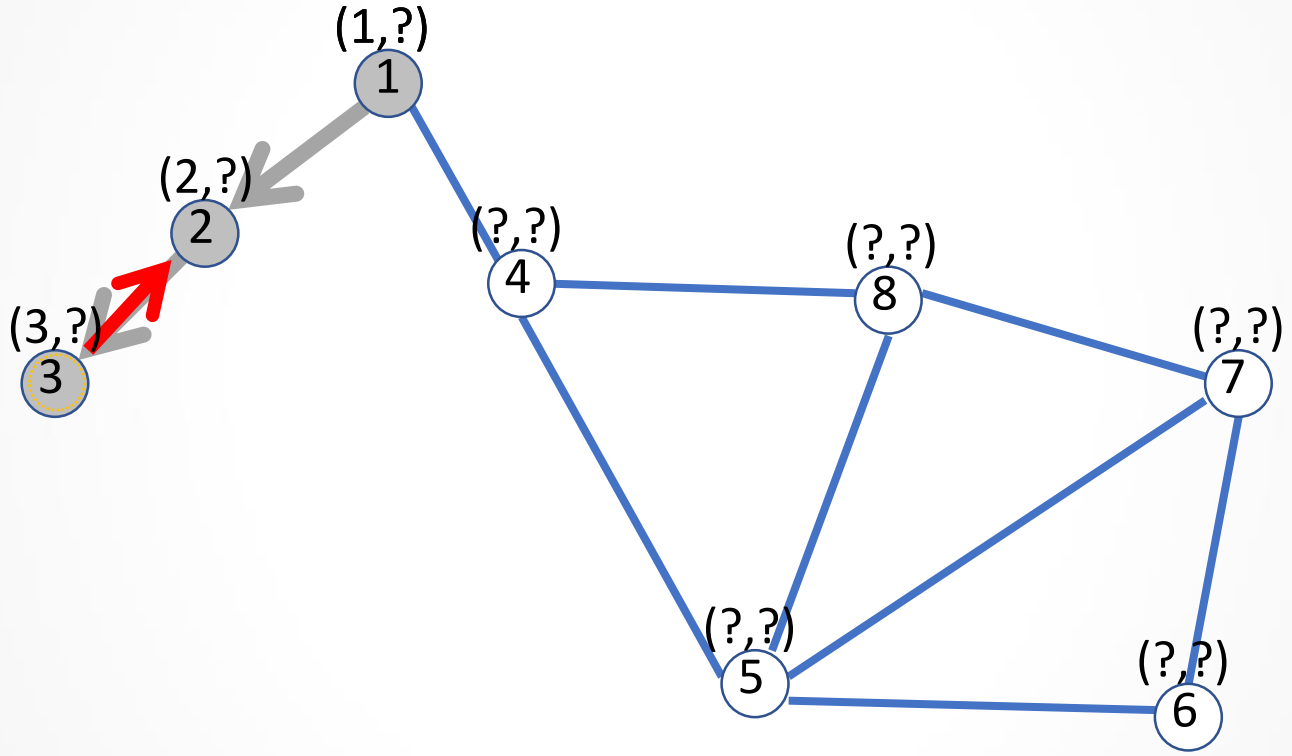
Обход в глубину: пример



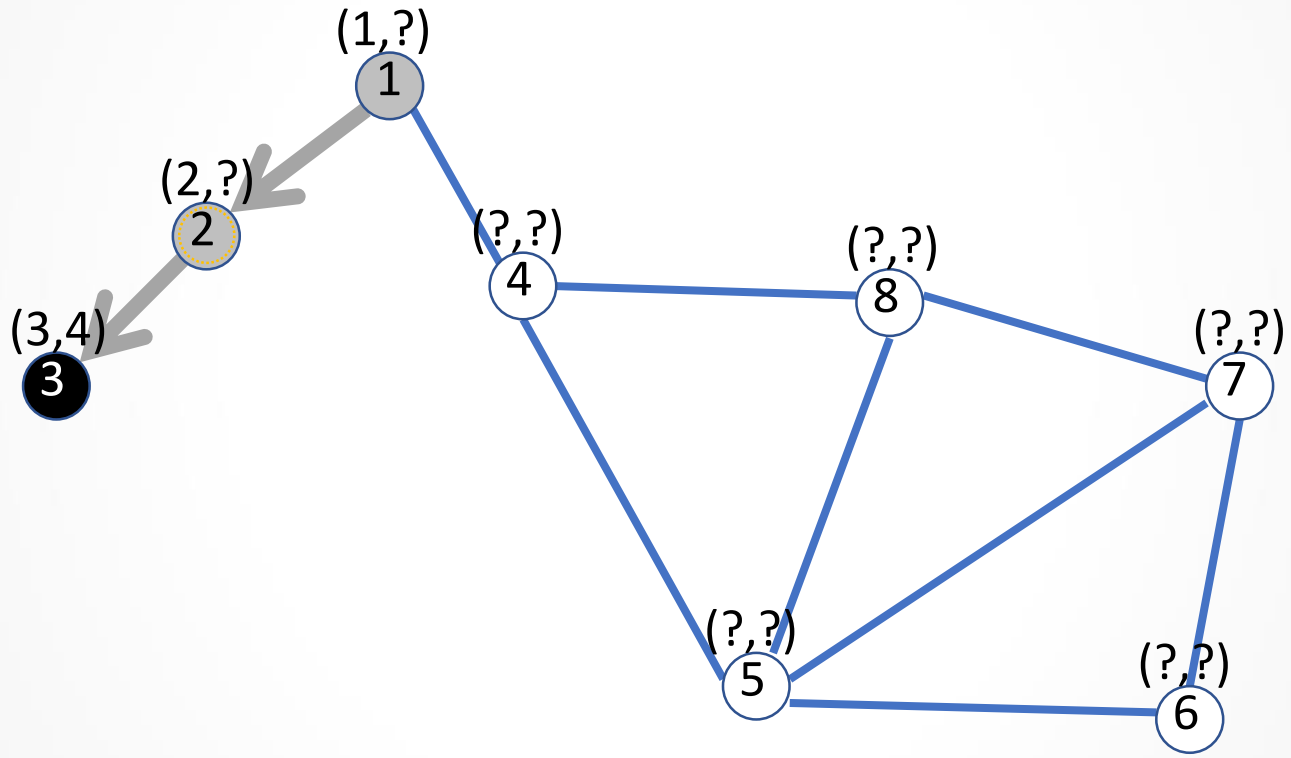
Обход в глубину: пример



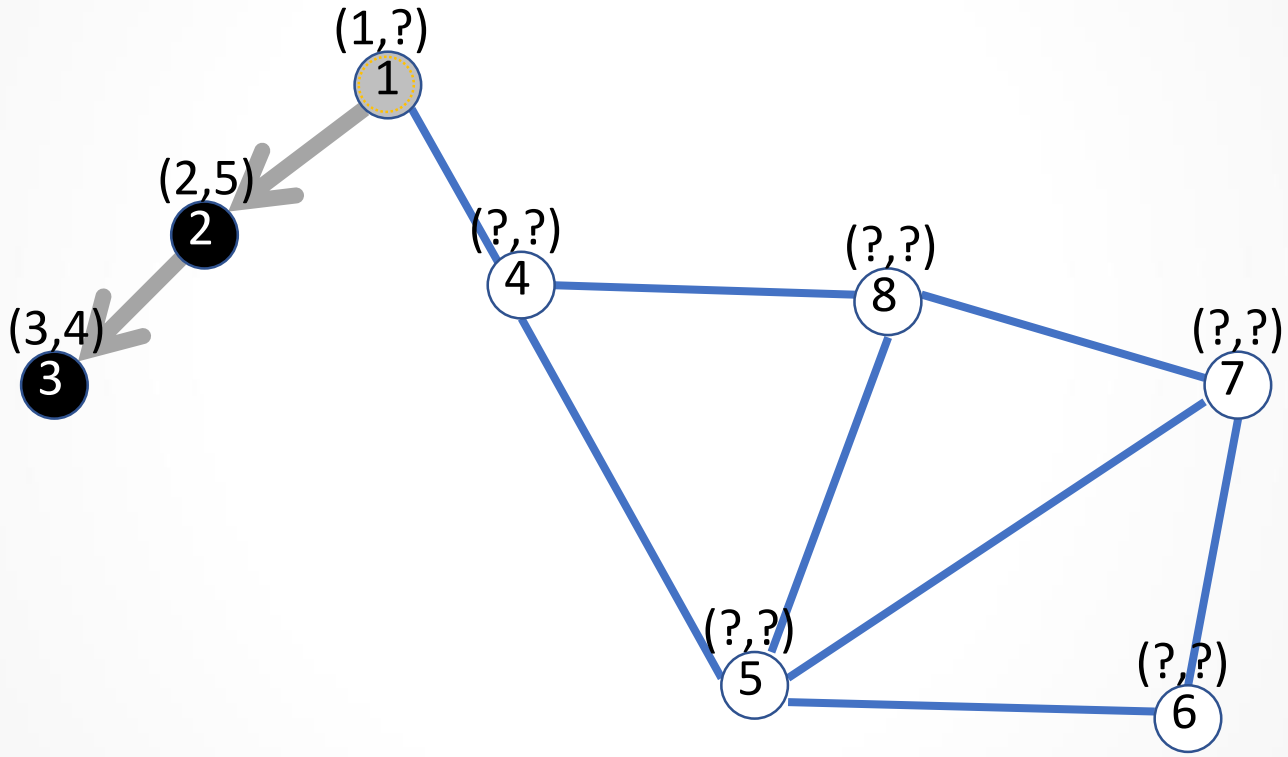
Обход в глубину: пример



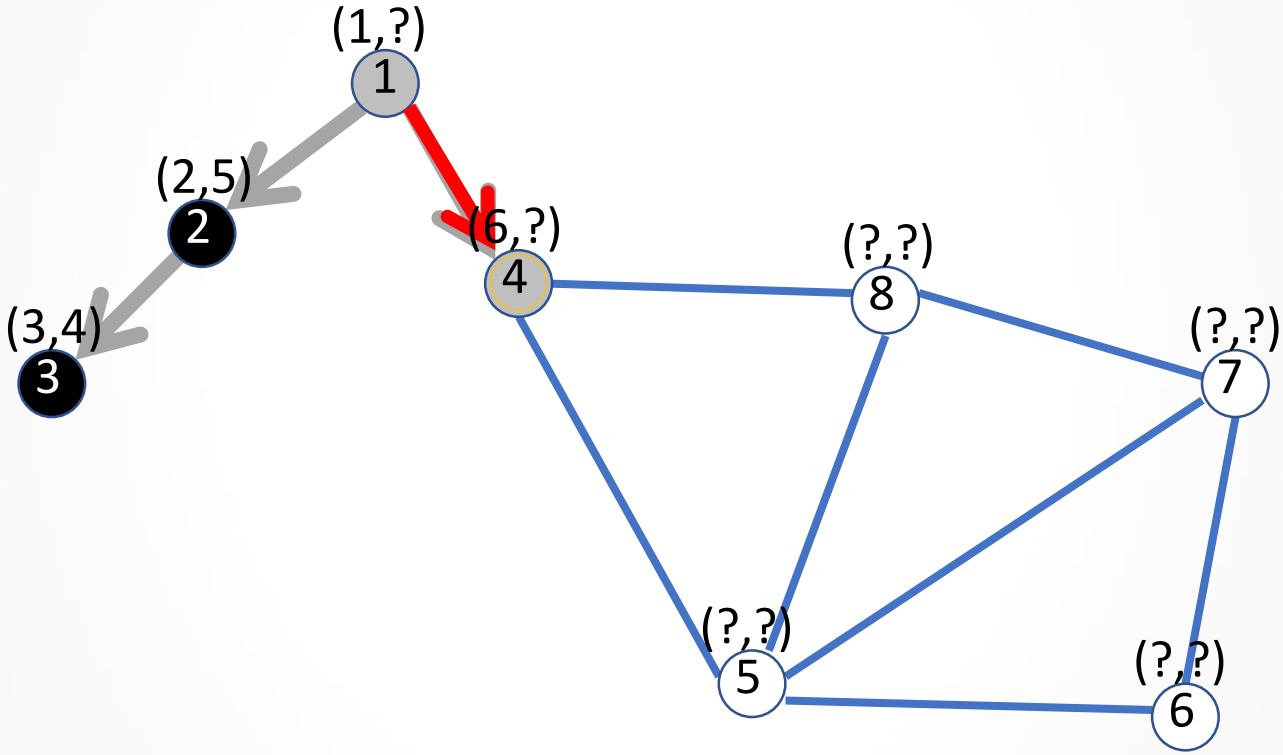
Обход в глубину: пример



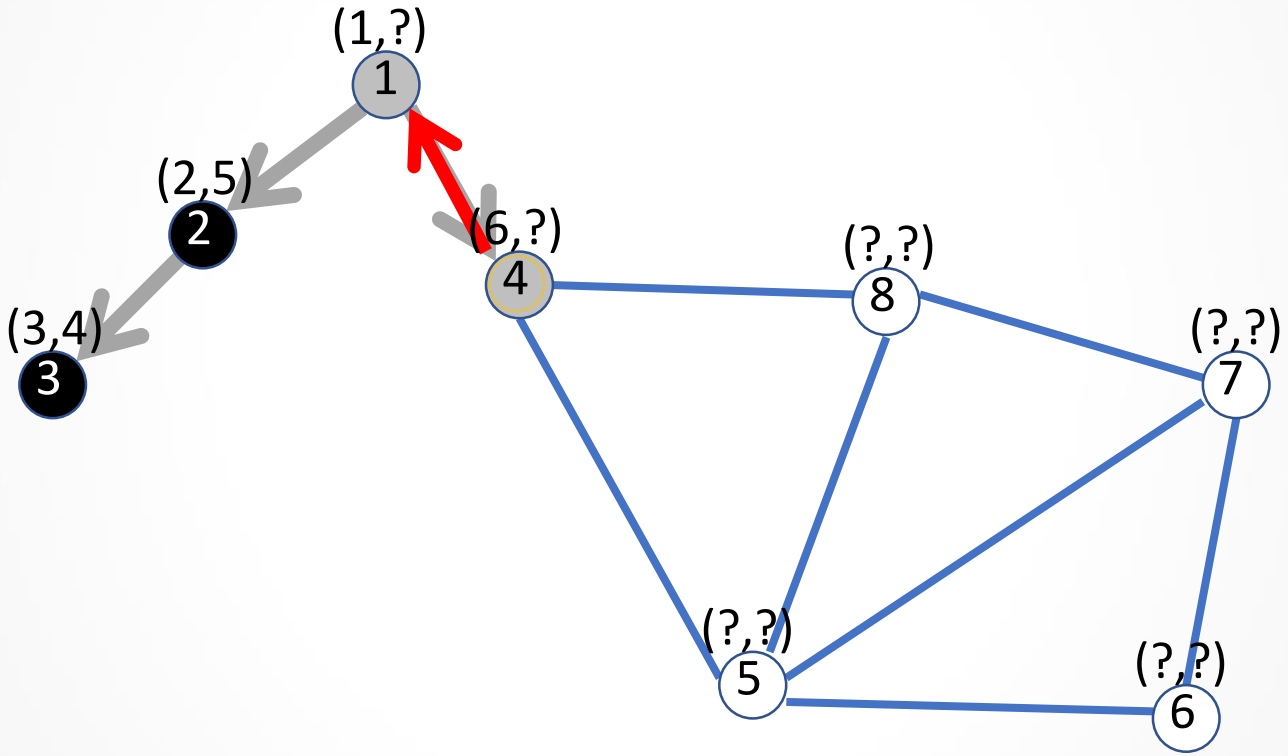
Обход в глубину: пример



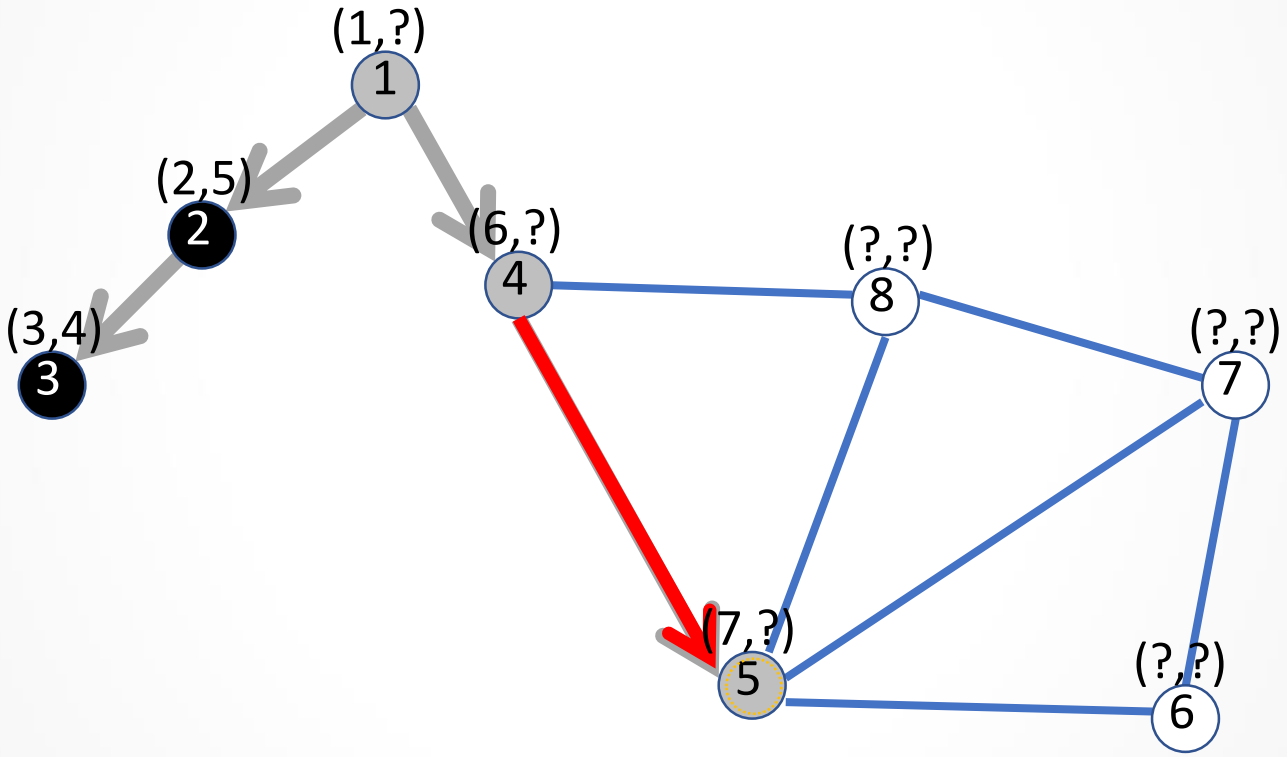
Обход в глубину: пример



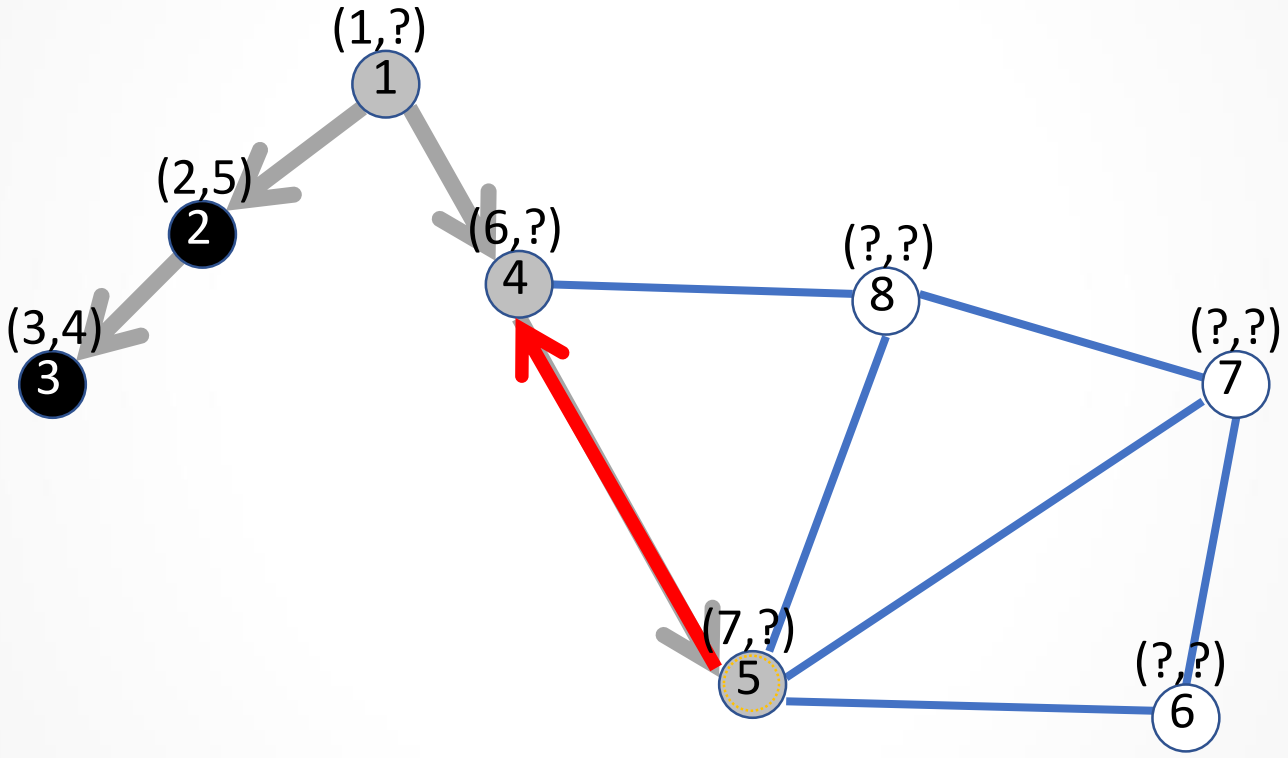
Обход в глубину: пример



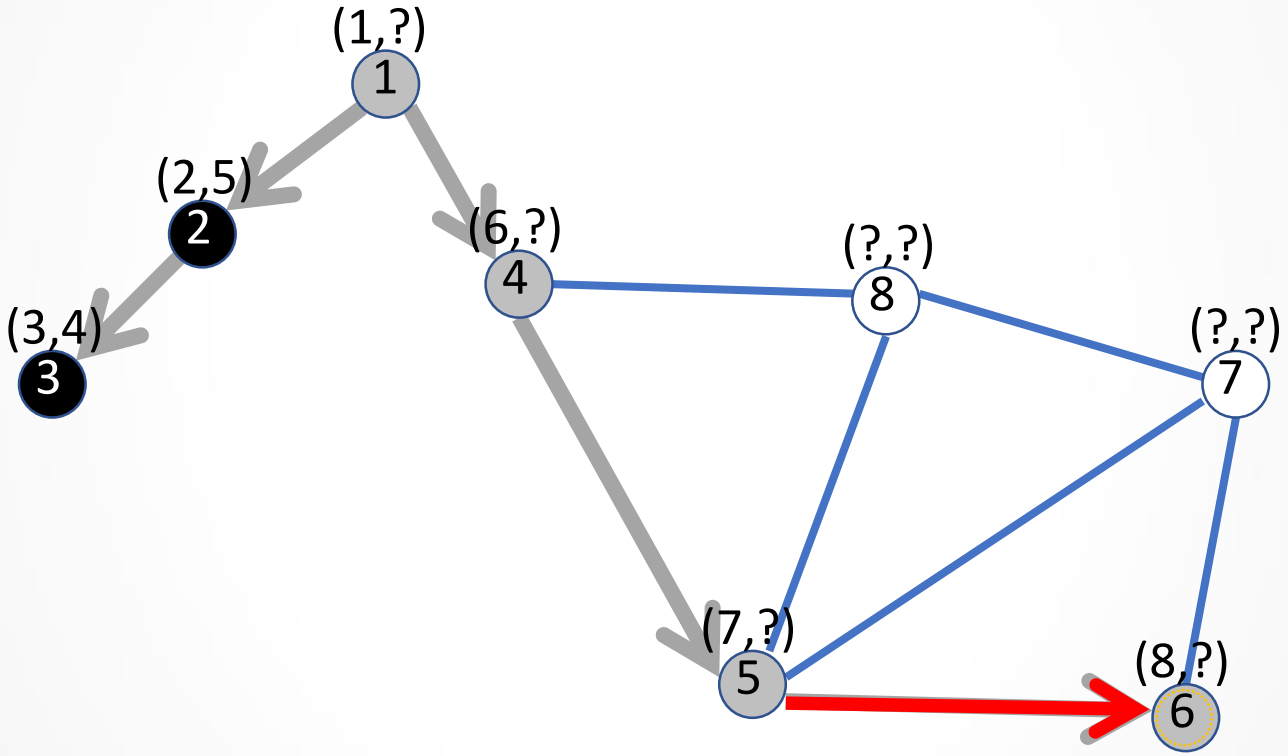
Обход в глубину: пример



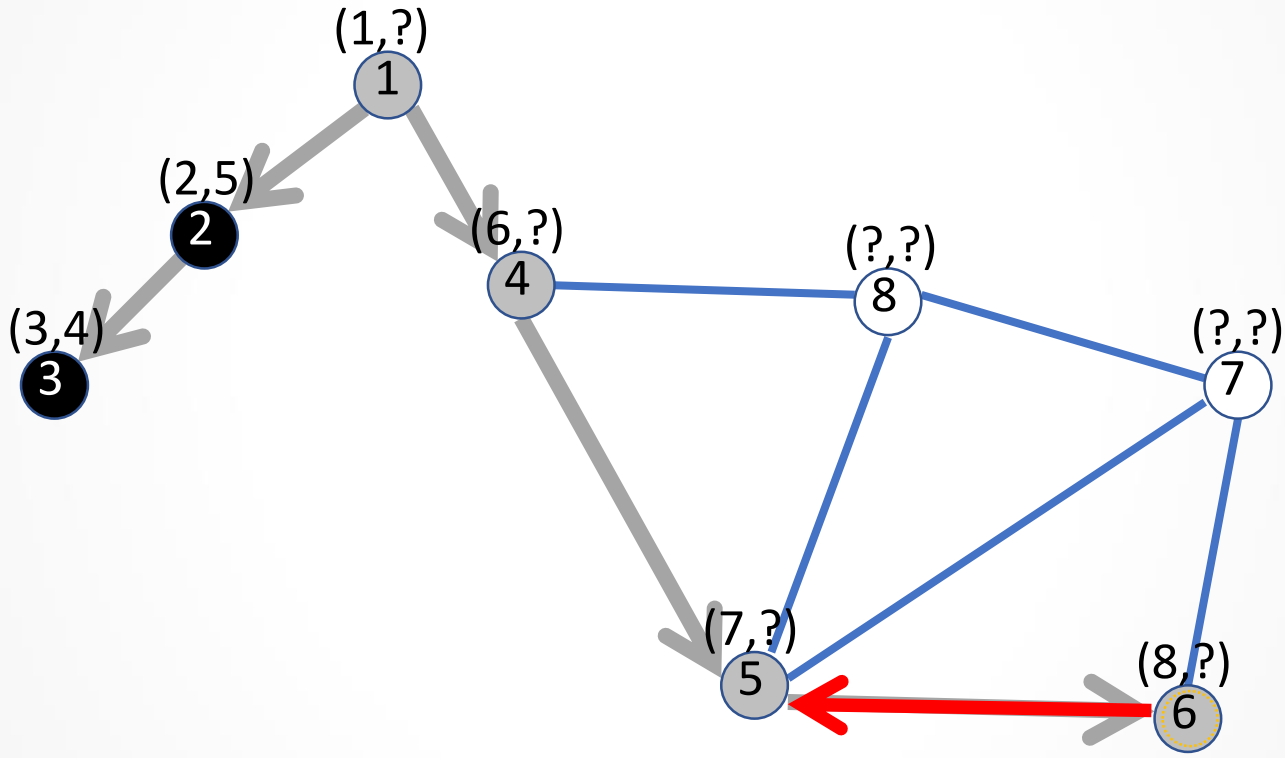
Обход в глубину: пример



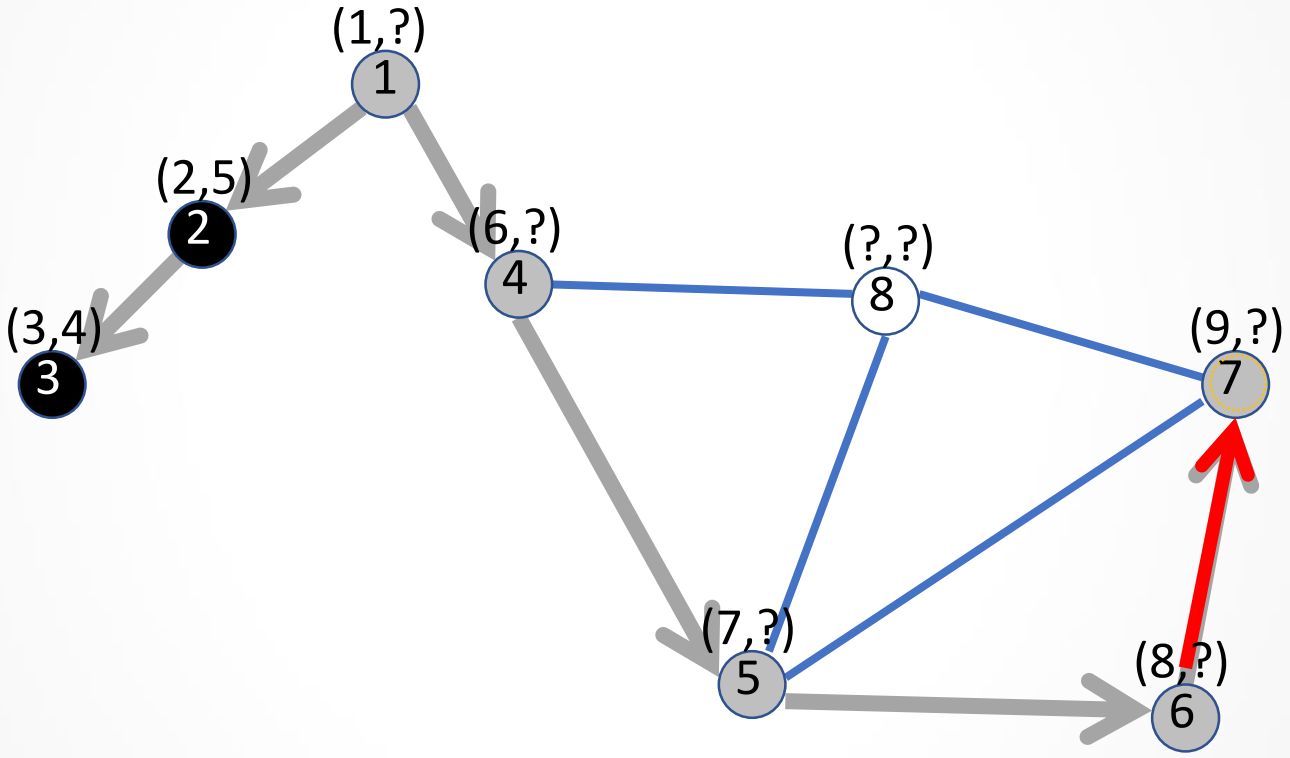
Обход в глубину: пример



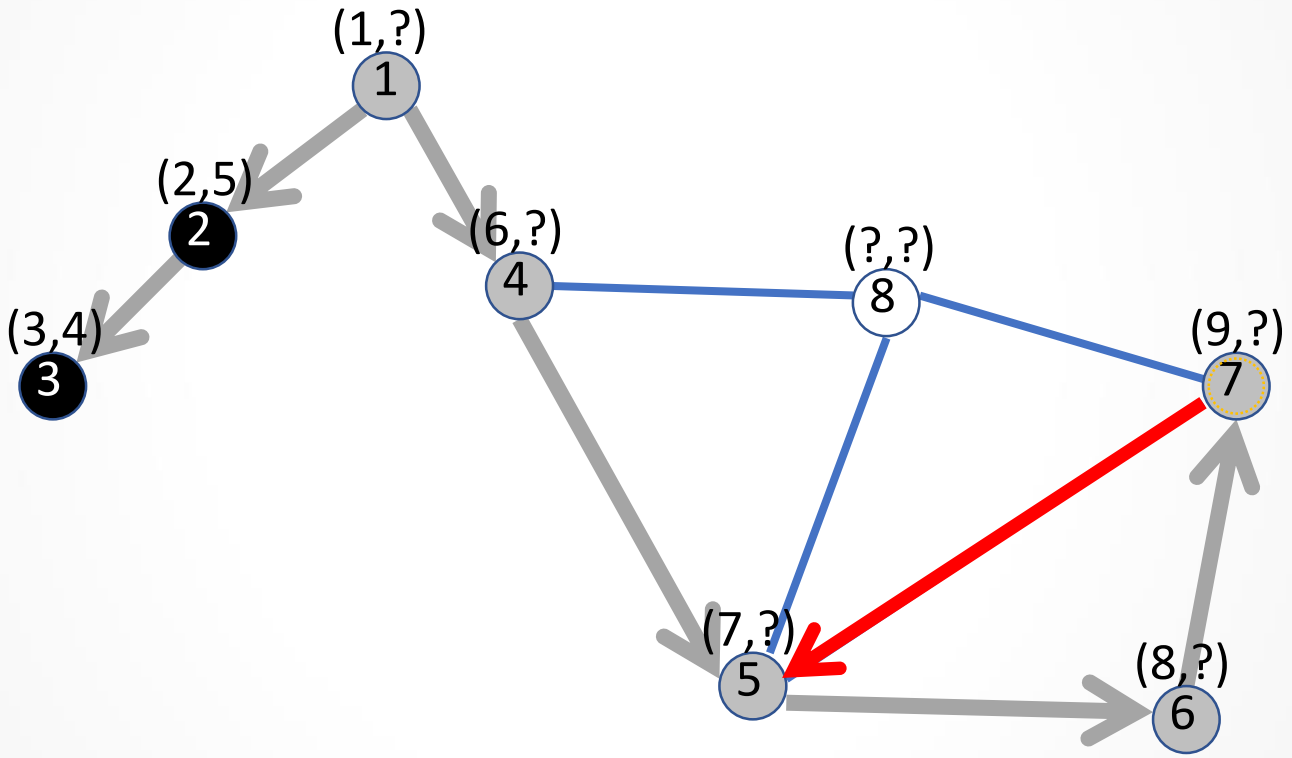
Обход в глубину: пример



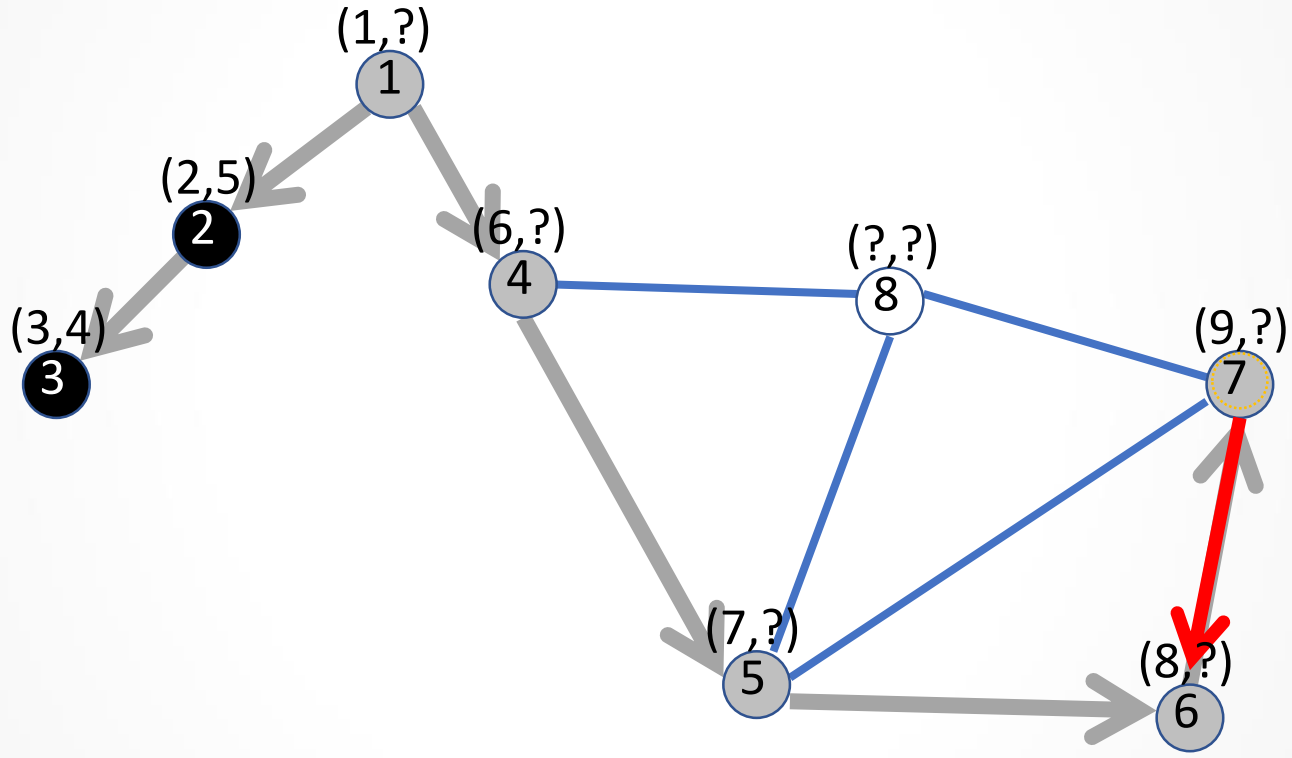
Обход в глубину: пример



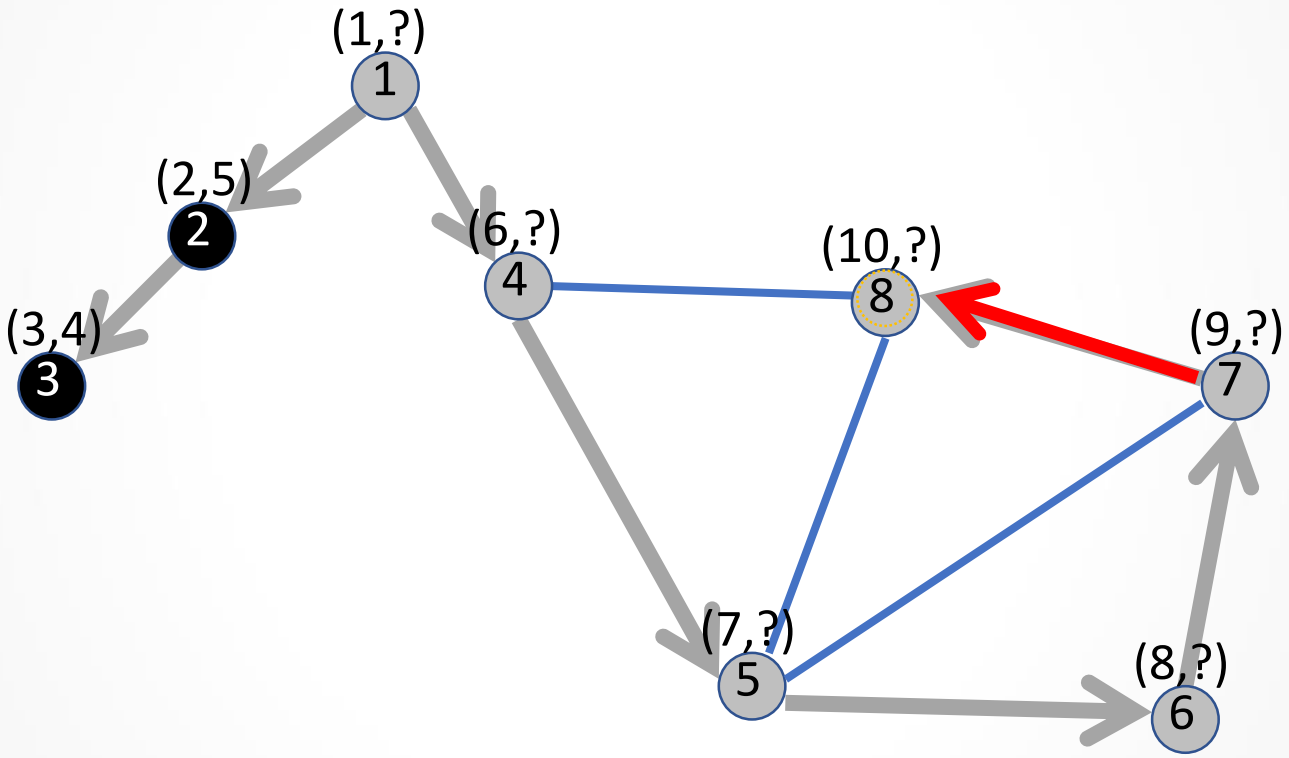
Обход в глубину: пример



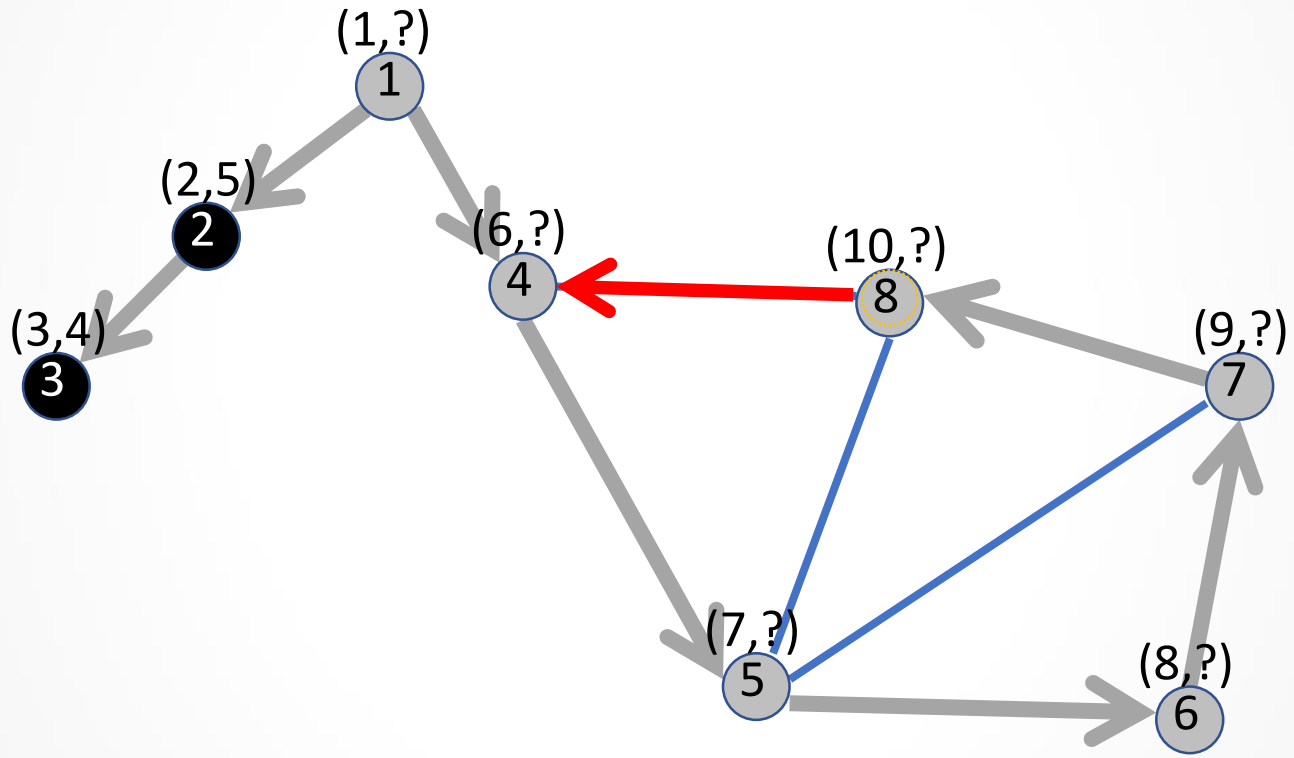
Обход в глубину: пример



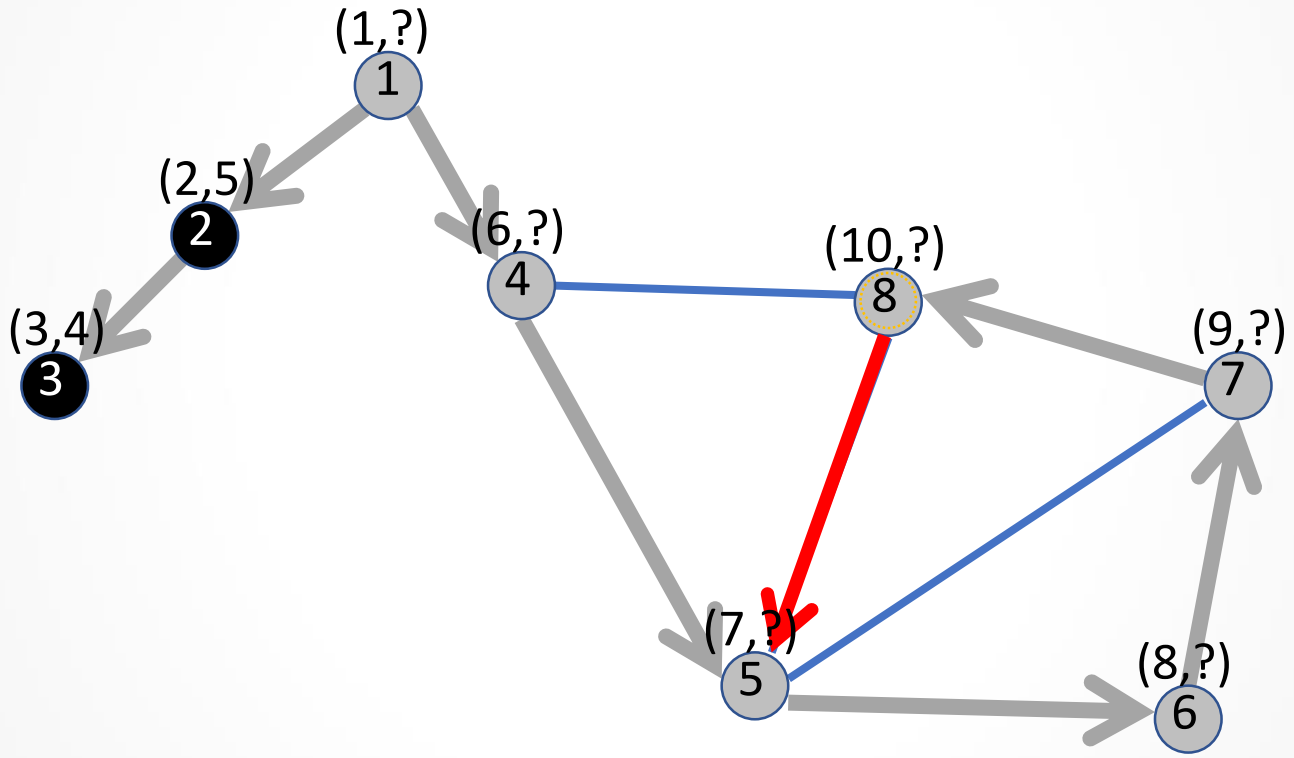
Обход в глубину: пример



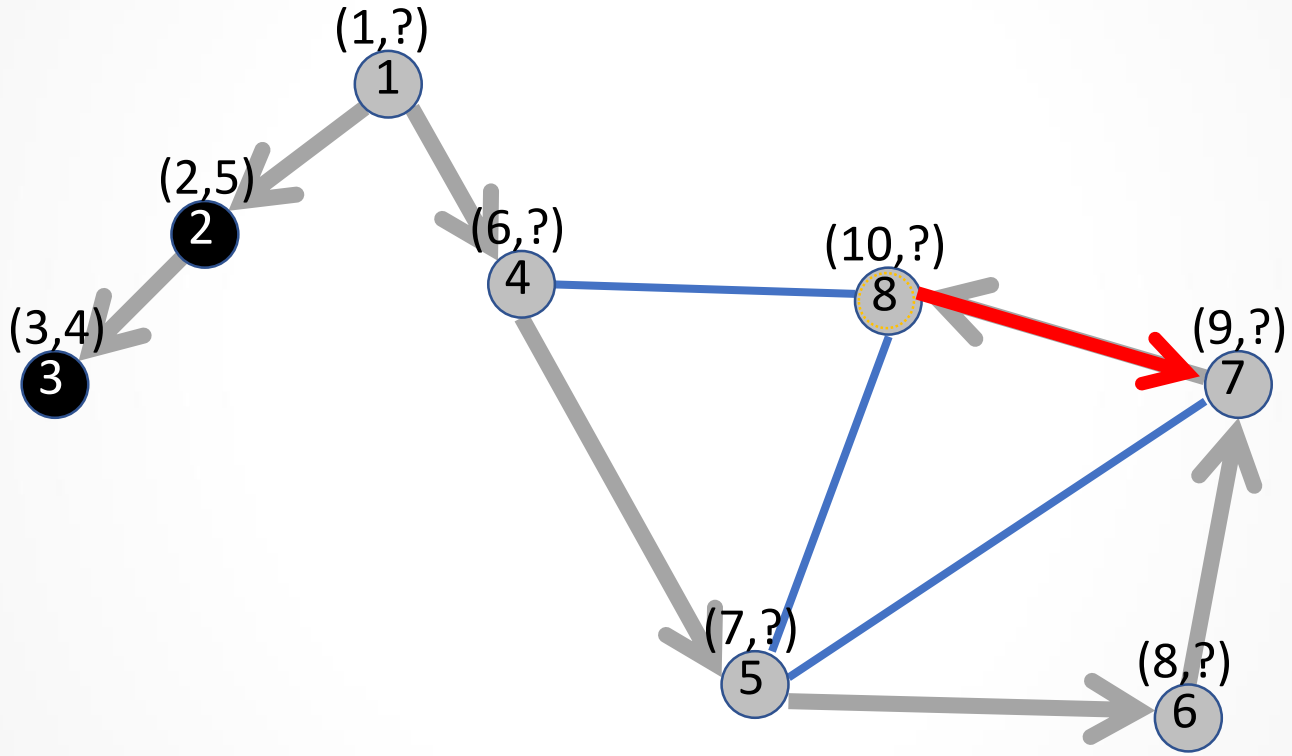
Обход в глубину: пример



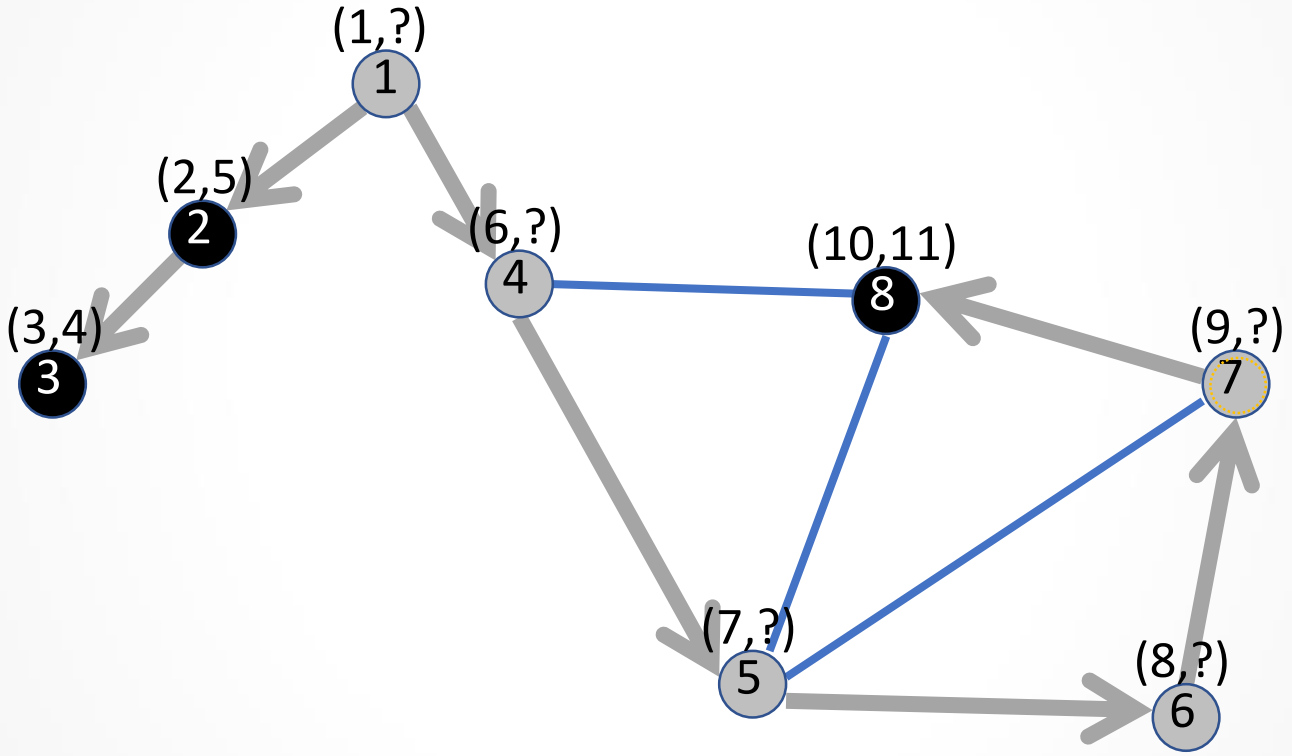
Обход в глубину: пример



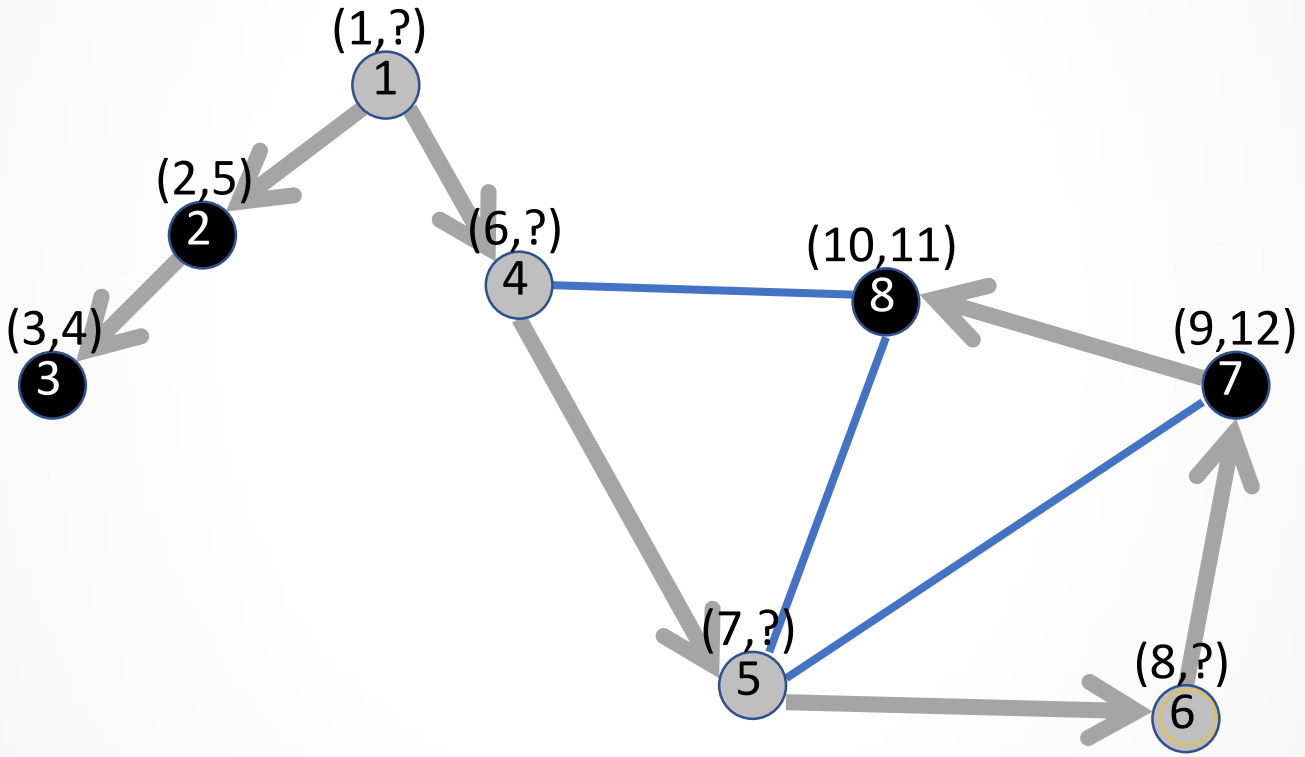
Обход в глубину: пример



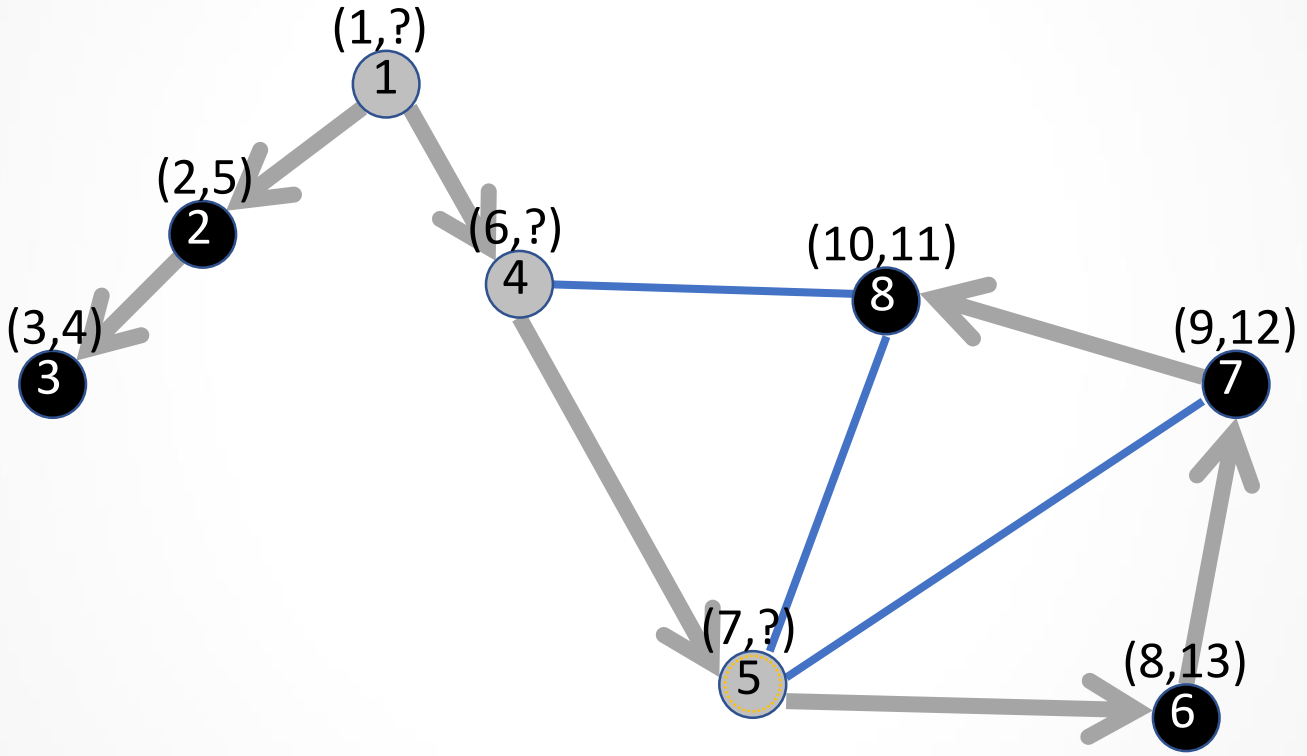
Обход в глубину: пример



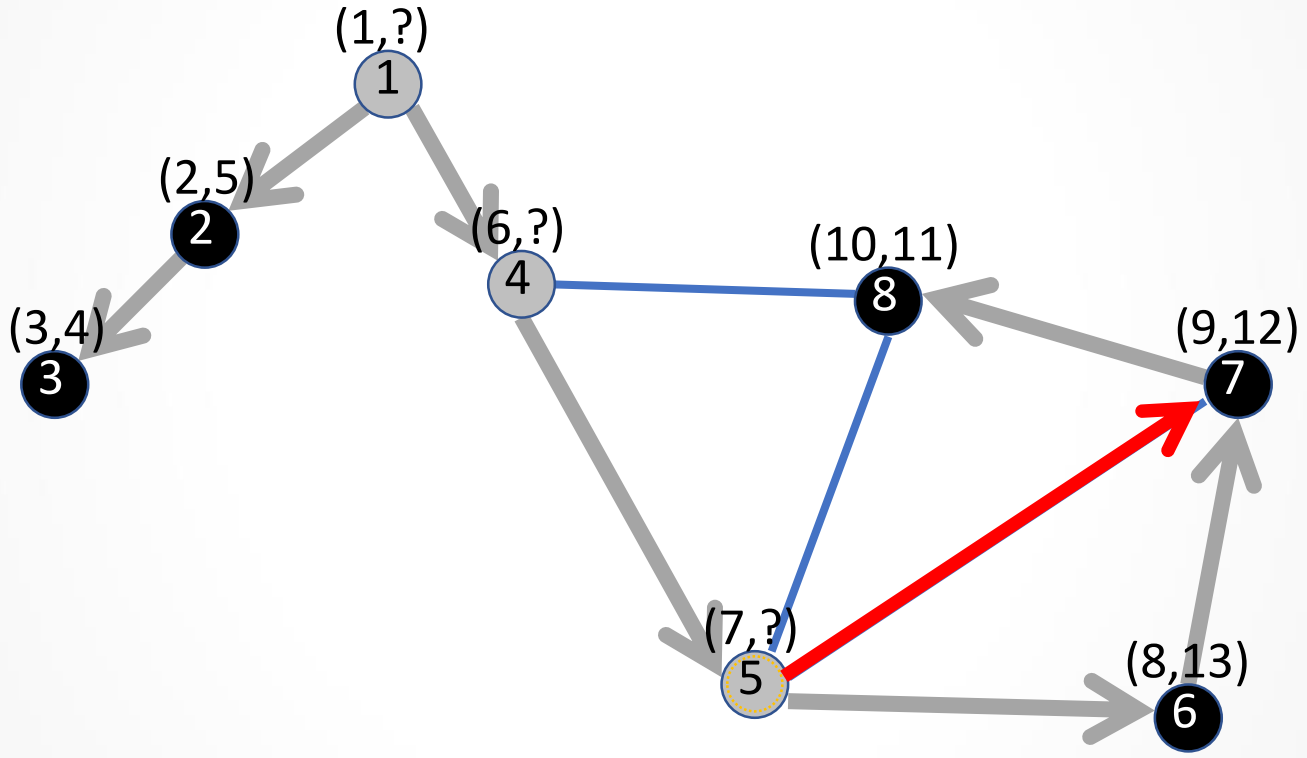
Обход в глубину: пример



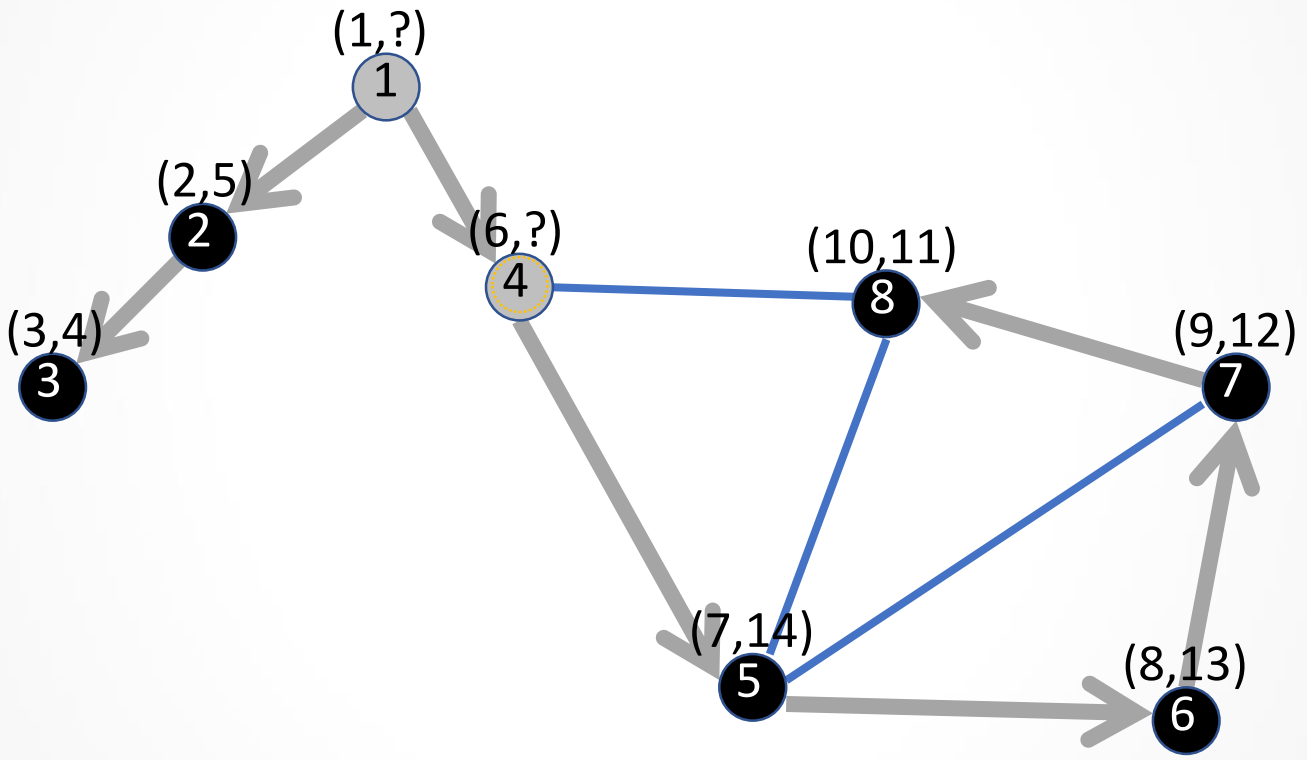
Обход в глубину: пример



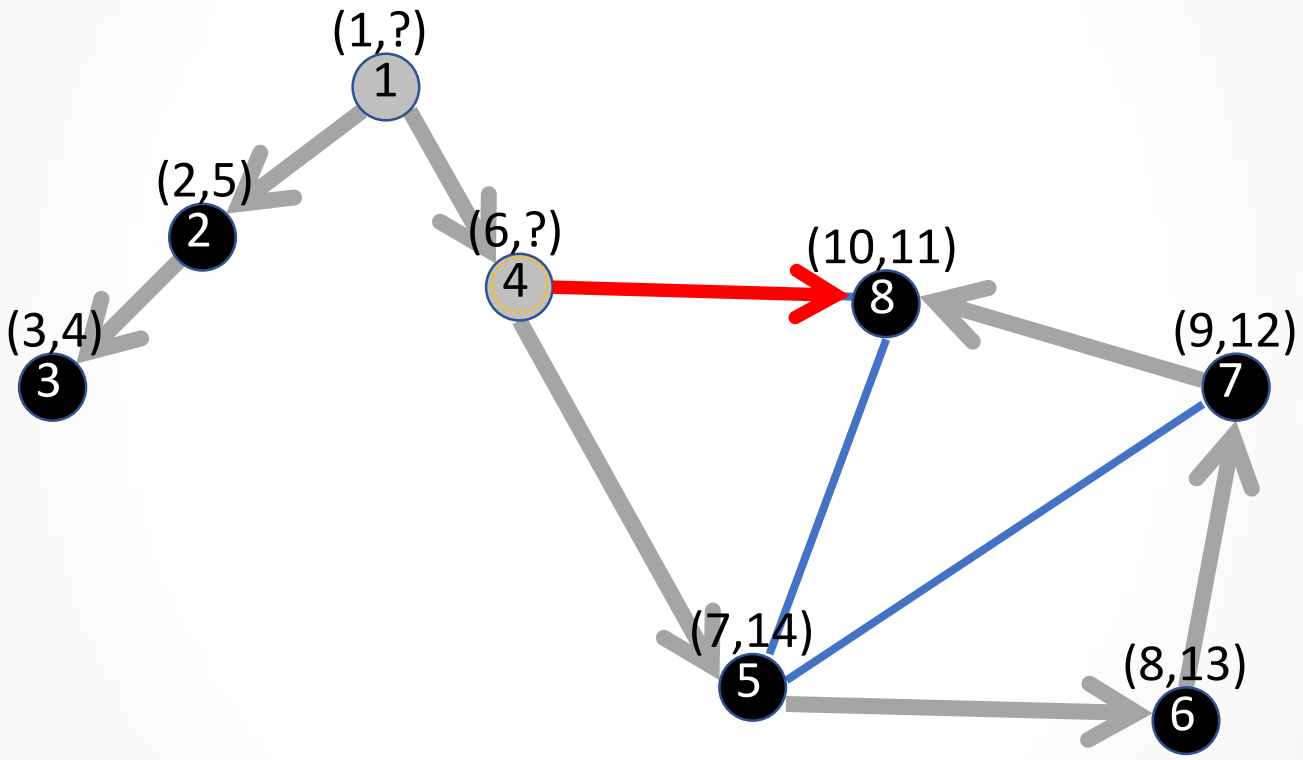
Обход в глубину: пример



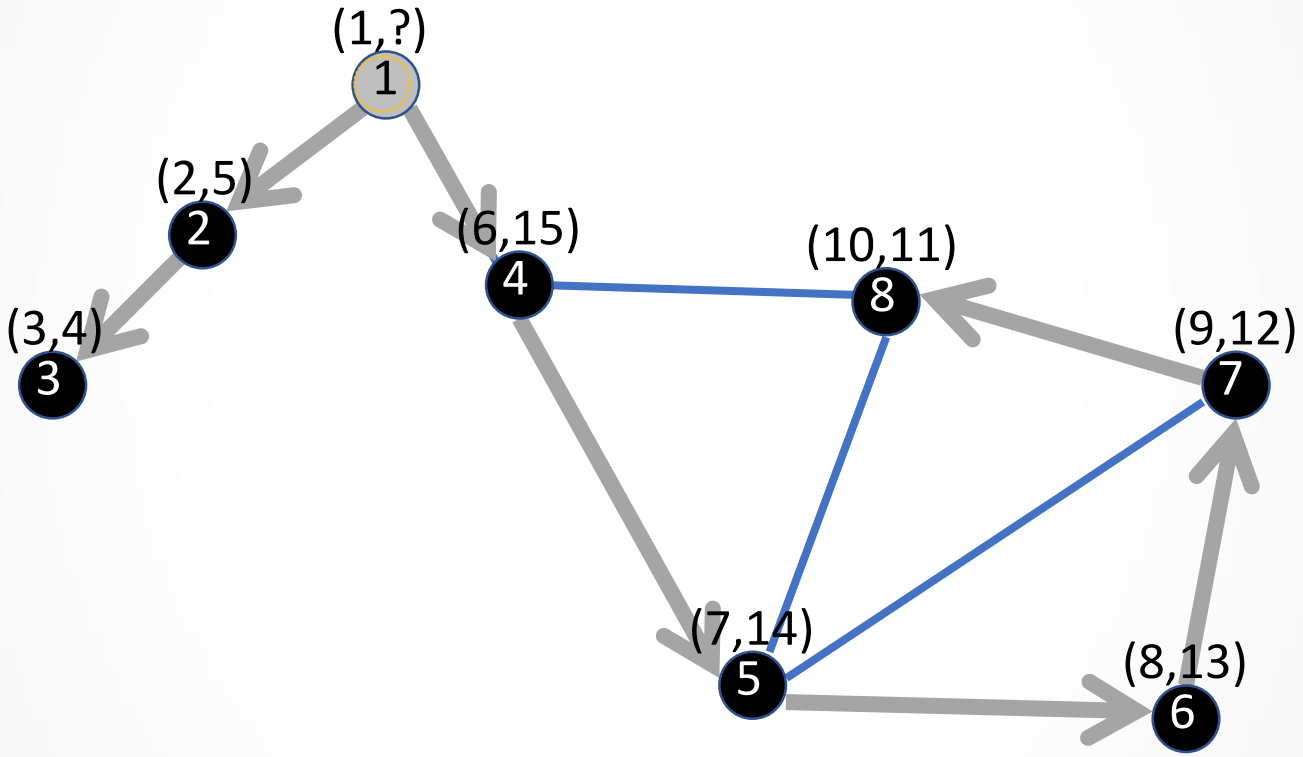
Обход в глубину: пример



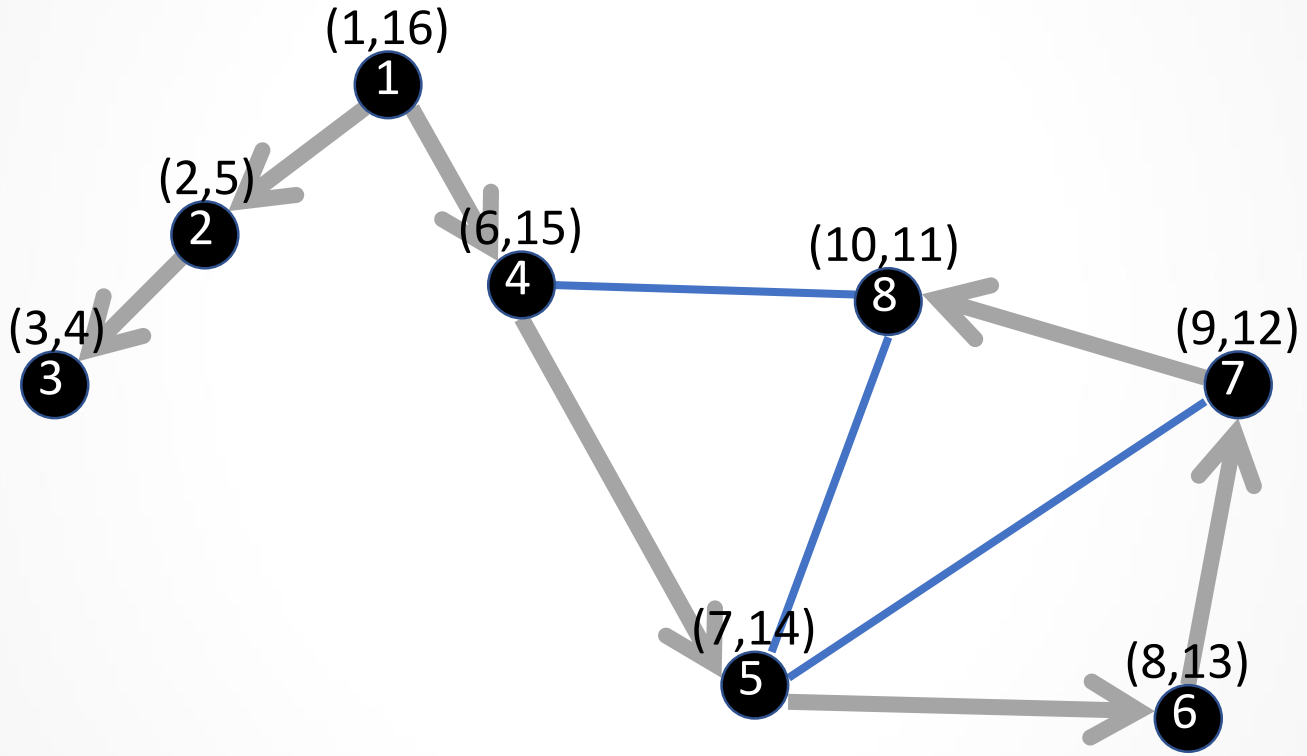
Обход в глубину: пример



Обход в глубину: пример



Обход в глубину: пример



Алгоритм обхода в глубину

- Для учёта правильного порядка обхода вершин будем применять стек или рекурсию.
- Для обхода графа в глубину нужны такие структуры данных:
  - ✓ Для учёта статуса вершины – массив `State[]`.
  - ✓ Сохранение предшественников вершин в массиве `Pred[]`.
  - ✓ Для учёта времён входа и выхода в вершины – массивы `Time_In[]` и `Time_Out[]`.

Алгоритм обхода в глубину

```
DFS(G)
For each v ∈ V:
   State[v] := 'unvisited';
   Pred[v] := NULL;
   Time_In[v] := NULL;
   Time_Out[v] := NULL;
CurTime : = 0;
For each v ∈ V:
   If State[v] = 'unvisited'
       DFS_Visit(v);
```

Алгоритм обхода в глубину

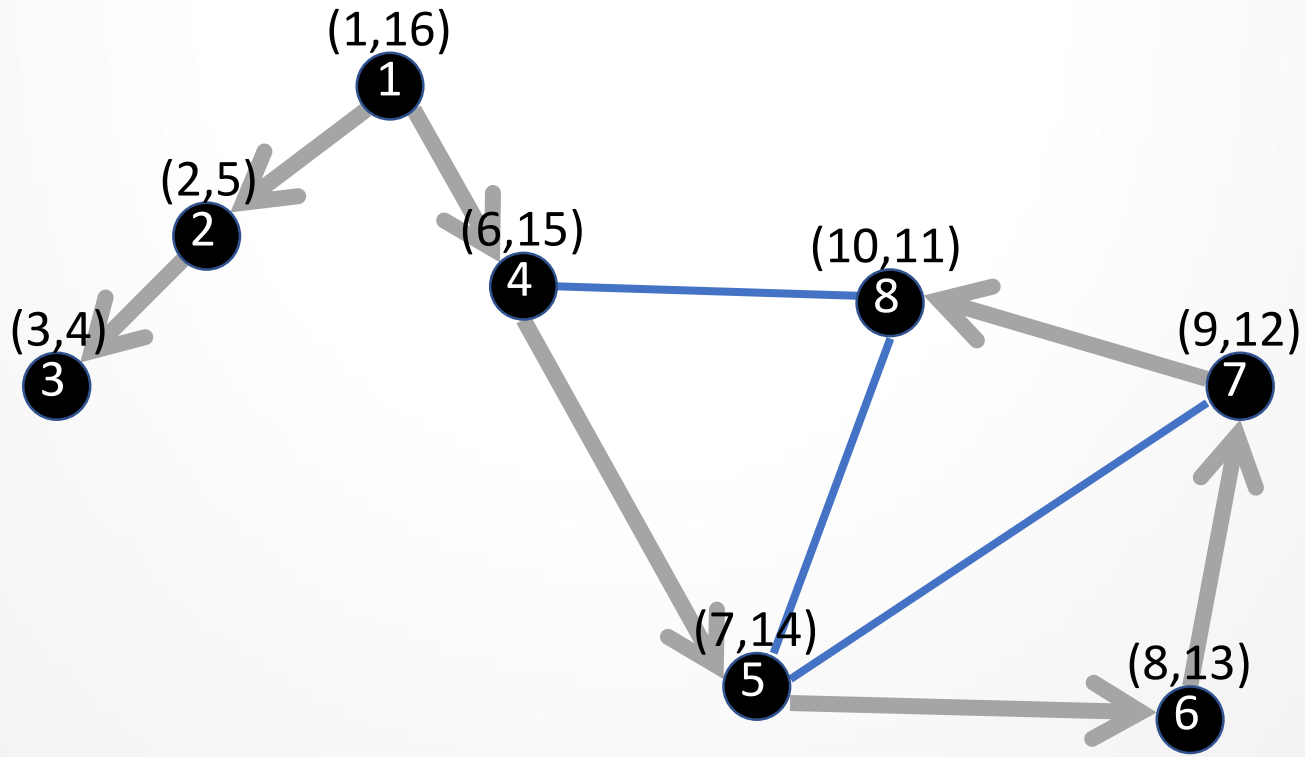
```
DFS_Visit(v)
State[v] := 'visited';
CurTime := CurTime + 1;
Time_In[v] := CurTime;
For each u in Adj(v)
   If State[u] = 'unvisited'
        Pred[u] := v;
        DFS_Visit(u);
State[v] := 'processed';
CurTime := CurTime + 1;
Time_Out[v] := CurTime;
```

Алгоритм обхода в глубину

Полезные свойства пометок `Time_In[]` и `Time_Out[]`.

1. Если вершина *x* – предшественник (не обязательно непосредственный) вершины *y*, то

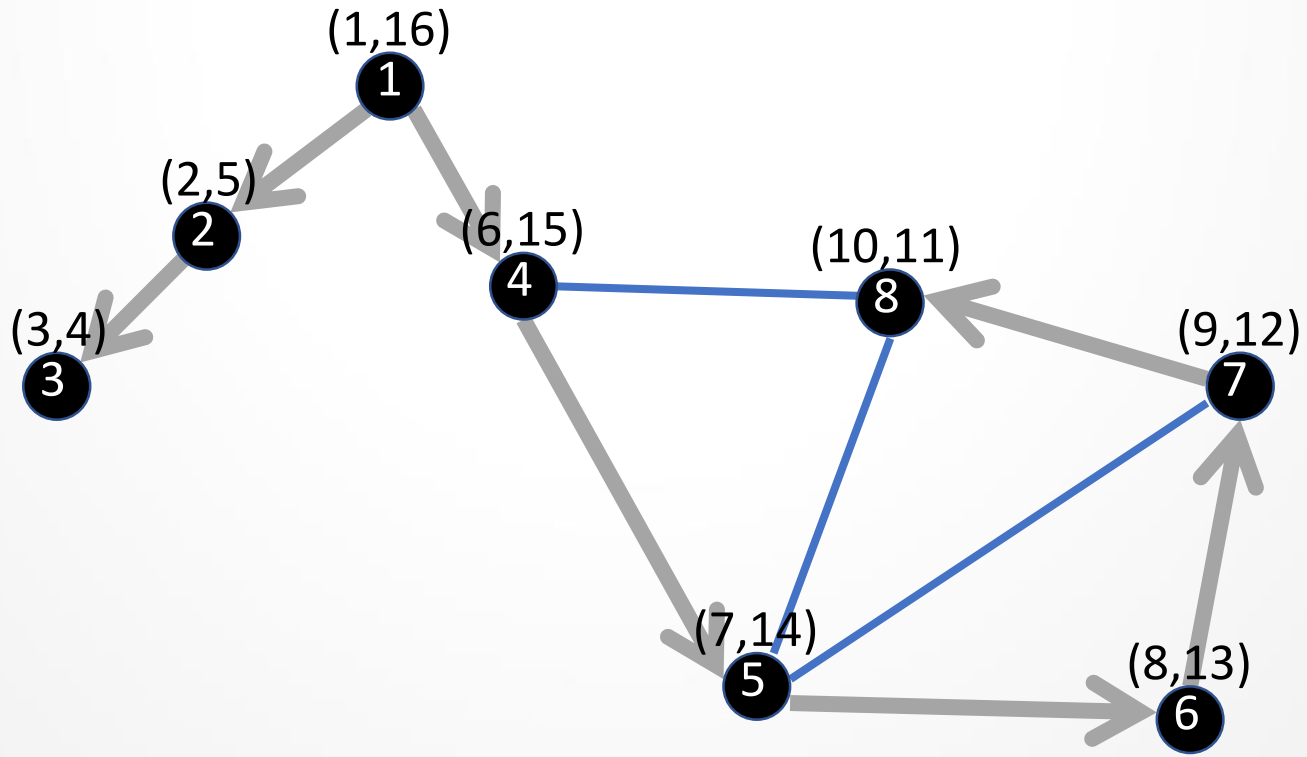
   Time_In[x] < Time_In[y] < Time_Out[y] < Time_Out[x]



Алгоритм обхода в глубину

Полезные свойства пометок `Time_In[]` и `Time_Out[]`.

2. Величина (Time_Out[x] - Time_In[x] - 1)/2 равна количеству древесных потомков вершины *x*.



Сложность обходов

Оценим временную сложность процедур обхода в ширину и в глубину.

При каждой процедуре обхода мы ровно по 2 раза обрабатываем каждую вершину и каждое ребро. Поэтому временная сложность составляет

$O(n + m)$.

Применение обхода в глубину

- ✓ *Поиск компонент связности*
- ✓ *Обнаружение циклов*
- ✓ Проверка двудольности графа
- ✓ Поиск мостов на графе

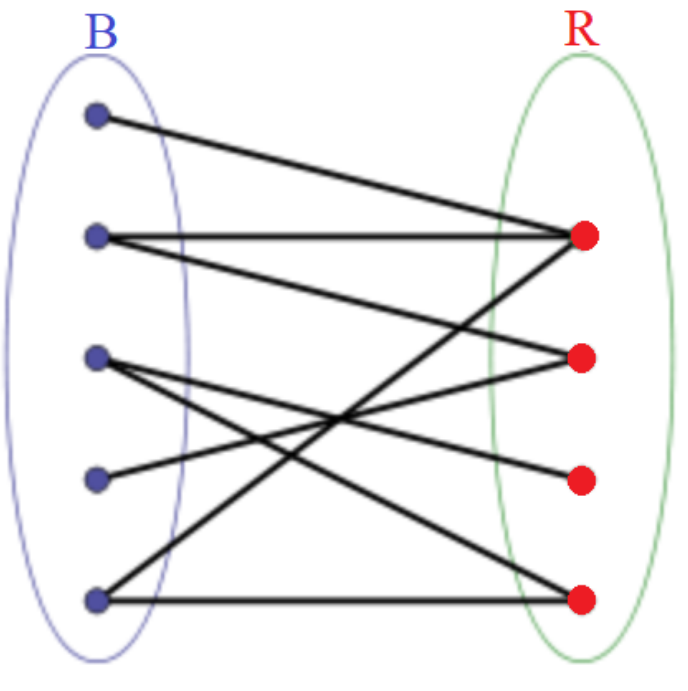
Применение обхода в глубину

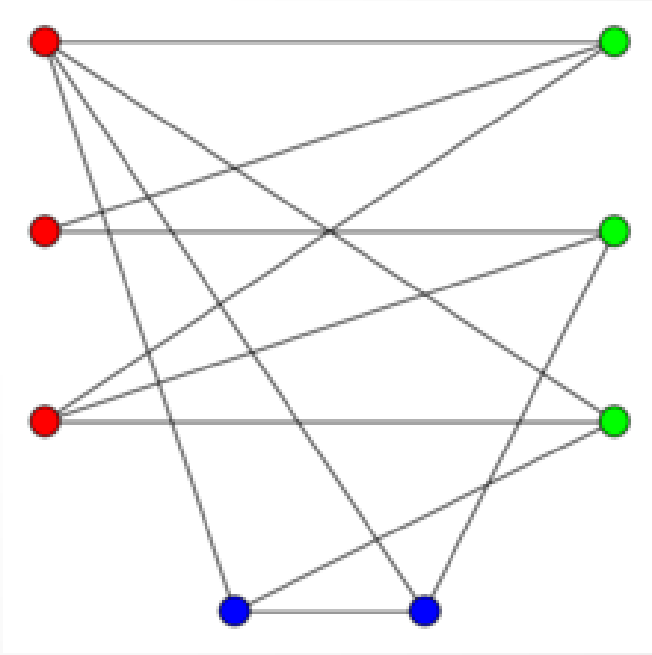
- ✓ *Поиск компонент связности*
- ✓ *Обнаружение циклов*

Для каждой из этих двух задач алгоритм на основе обхода в глубину по своей сути совпадает с аналогичным алгоритмом, основанным на обходе в ширину.

Двудольность

Определение. Граф $G(V, E)$ называется ***двудольным***, если множество вершин можно разбить на два непересекающиеся подмножества $V = B \cup R$ таким образом, что концы каждого ребра лежат в разных подмножествах.

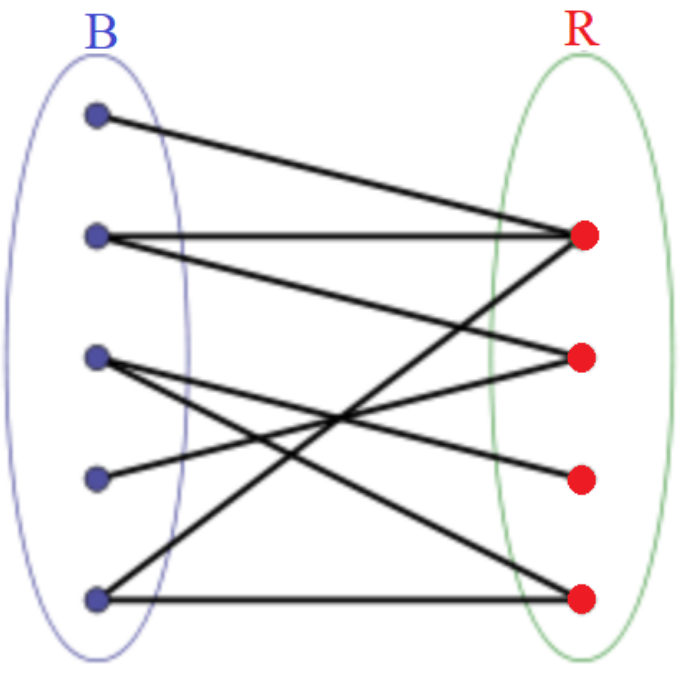


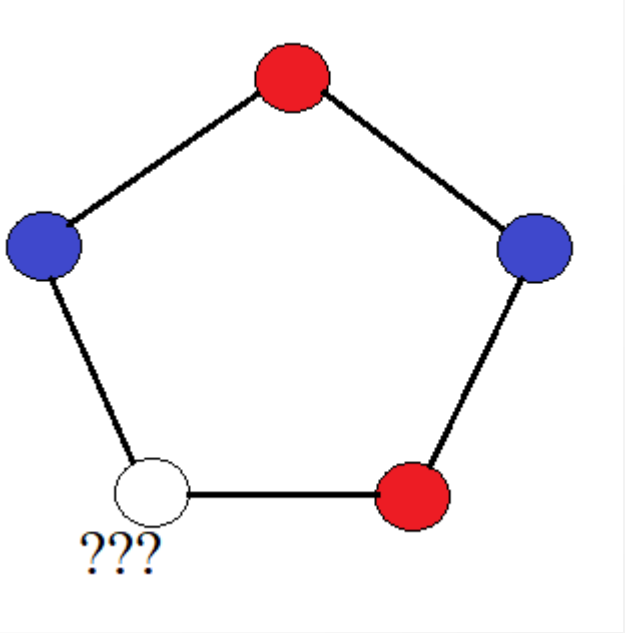


Двудольность

Теорема. Граф $G(V, E)$ является двудольным тогда и только тогда, когда на нём нет циклов нечётной длины.

Следствие. Деревья и леса – двудольные графы.

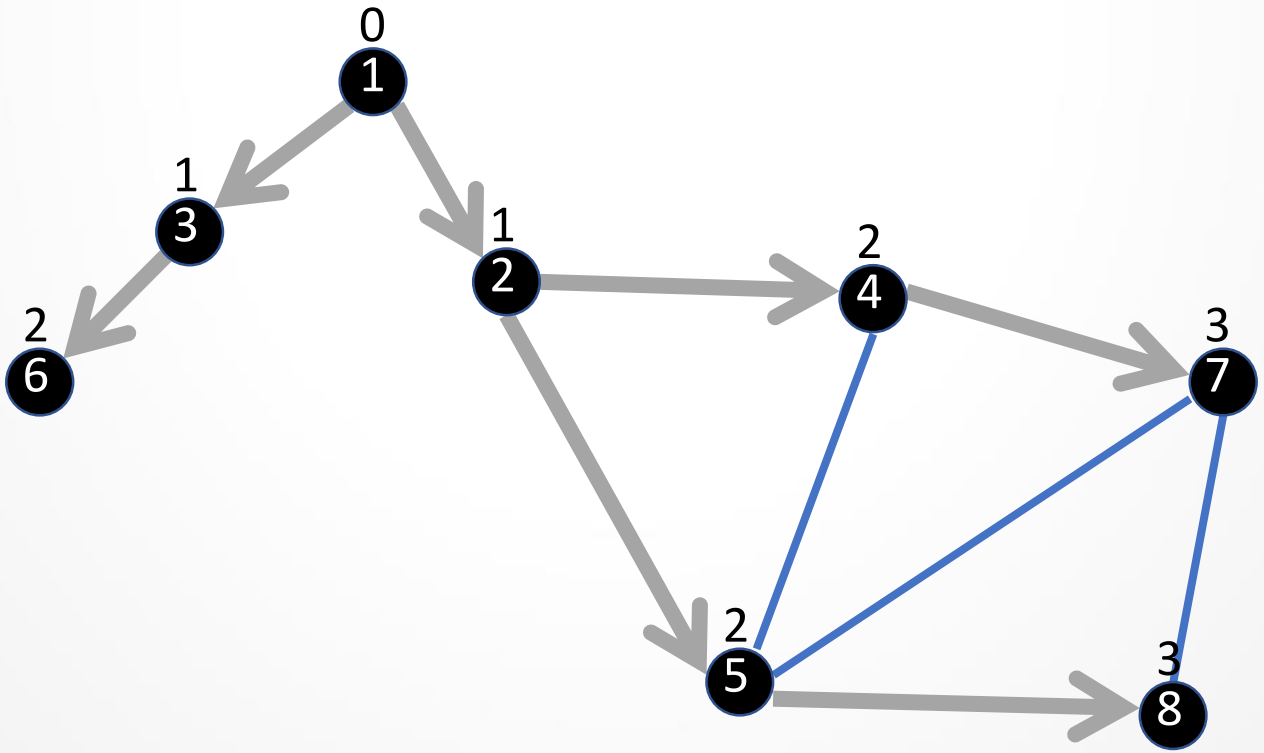




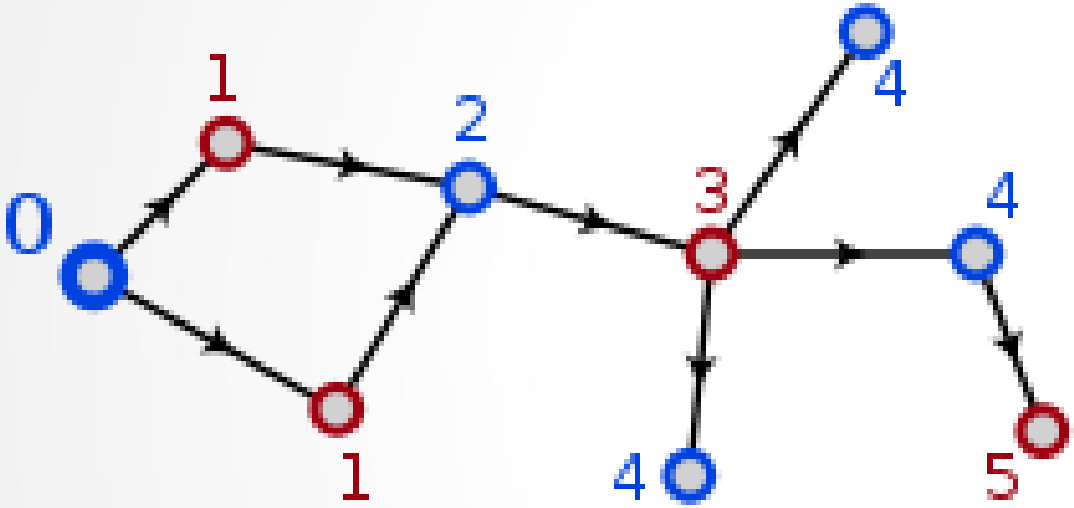
Двудольность

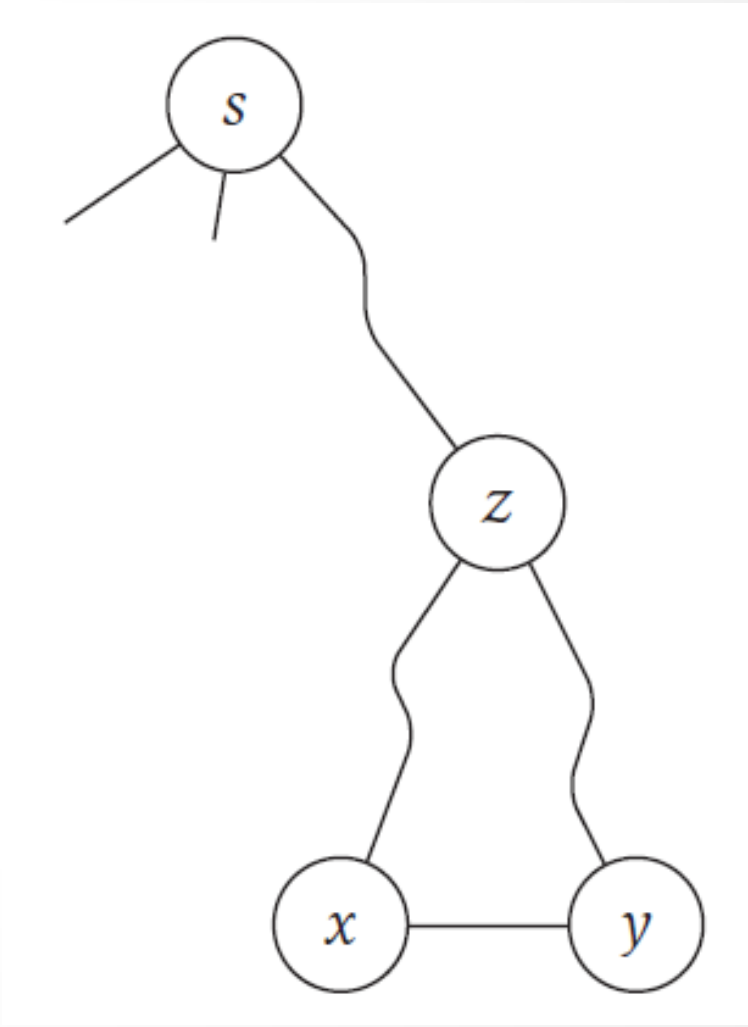
Проверка двудольности с помощью обхода в ширину.

1. Рассчитаем расстояния Dist[] от начальной вершины до всех остальных.
2. Любой цикл нечётной длины соединяет вершины *u* и *v*, для которых `Dist[u]` и `Dist[v]` имеют одинаковую чётность.



Двудольность





Двудольность

Алгоритм проверки двудольности с помощью обхода в ширину.

Пусть $G(V, E)$ - связный граф.

1. $R = B = \emptyset$
2. Выбрать произвольную вершину $s \in V$. Dist[s]=0.
3. Рассчитать Dist[] – расстояния от s до всех остальных вершин.
4. For each $v \in V$:
   if Dist[v] – нечётное: $R = R \cup \{v\}$
   else: $B = B \cup \{v\}$
5. Для всех рёбер проверить выполнение условия.

Временная сложность: $O(n + m)$.

Двудольность

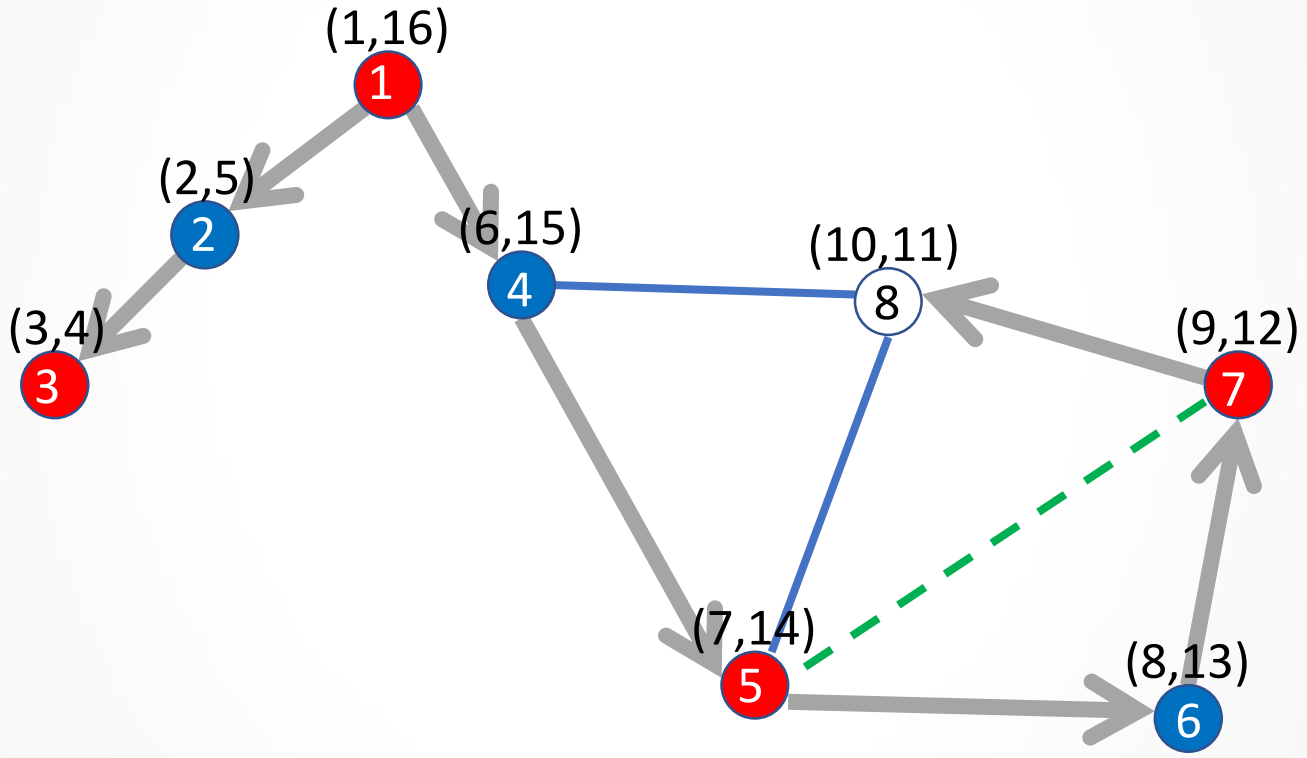
Проверка двудольности с помощью обхода в глубину.

В процессе обхода графа в глубину, разделяем вершины на две доли (*R* и *B*):

1. Выбираем произвольную начальную вершину и помещаем её в R.
2. Каждую новую вершину *u* помещаем в долю, отличную от доли, в которой находится её непосредственный предшественник Pred[u].
3. Если рассматриваемое ребро (v, u) ведёт в посещённую вершину, то проверяем, что *u* и *v* лежат в разных долях.

Временная сложность: $O(n + m)$.

Обход в глубину: пример



Практические задачи

Задача «Округа»

Собрался к царю Митрофану боярин Митяй на доклад. А доклад царь требовал о дорогах между *Загорским* и *Догорским* округом, проходящих через Бескрайние горы.

Боярин Митяй разузнал - какие города в тех округах и о дорогах, проходящих через горы. Да вот города у него перепутались ☹. А вот список дорог остался. Только и помнит Митяй, что дороги между городами одного округа он тщательно повычеркивал.

Помогите Митяю разобраться, а то не сносить ему головы.

Практические задачи

Задача «Закольцованный водопровод»

В водопроводной системе города Лкиц возникло закольцовывание, что привело к некорректным показателям давления на разных участках водопровода. В результате подача воды производится не так, как запланировано проектировщиками.

Ваша задача: определить, какие участки водопровода образовали кольцо. Известно, что кольцо единственно.

Практические задачи

Задача «Закольцованный водопровод»

Формат входных данных

В первой строке M - число участков трубопровода.

В следующих M строках заданы участки в виде пар координат узлов трубопровода.

Формат выходных данных

В первой строке количество участков трубопровода, входящих в кольцо.

В следующих строках описание закольцованного участка в виде последовательности пар координат. В каждой строке координаты одного узла трубопровода.

Практические задачи

Задача «Закольцованный водопровод»

Пример

input.txt	output.txt
8 1 1 3 3 3 3 4 0 7 0 4 0 0 5 2 0 3 3 2 0 1 1 7 0 0 5 1 9 0 5 2 3	4 1 1 3 3 4 0 7 0