Ориентированный граф (орграф)

$$G = (V, E)$$

Ориентированный граф (орграф)

$$G = (V, E)$$

$$E \subset V \times V$$

Ориентированный граф (орграф)

$$G = (V, E)$$

$$E \subset V \times V$$

$$V = \{2, 3, 4, 6, 8\}$$

Ориентированный граф (орграф)

$$G = (V, E)$$

$$E \subset V \times V$$

$$V = \{2, 3, 4, 6, 8\}$$

$$E = \{(4, 2), (6, 2), (6, 3), (8, 2), (8, 4)\} = \{(a, b) \ : \ a \in V, \ b \in V, \ a\%b = 0\}$$

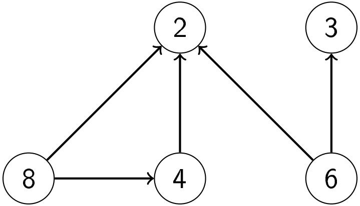
Ориентированный граф (орграф)

$$G = (V, E)$$

$$E \subset V \times V$$

$$V = \{2, 3, 4, 6, 8\}$$

$$E = \{(4, 2), (6, 2), (6, 3), (8, 2), (8, 4)\} = \{(a, b) \ : \ a \in V, \ b \in V, \ a\%b = 0\}$$



Неориентированный граф

$$G = (V, E)$$

Неориентированный граф

$$G = (V, E)$$

$$E = \{\ \{a, b\}\ :\ a, b \in V\}$$

Неориентированный граф

$$G = (V, E)$$

$$E = \{\ \{a, b\}\ :\ a, b \in V\}$$

$$V = \{1, 2, 3, 4, 5\}$$

Неориентированный граф

$$G = (V, E)$$

$$E = \{\ \{a, b\}\ :\ a, b \in V\}$$

$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(1, 3), (1, 5), (3, 5), (2, 4)\} = \{(a, b)\ :\ a \in V,\ b \in V,\ (a + b)\%2 = 0\}$$

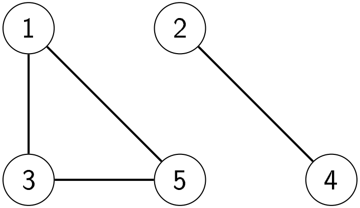
Неориентированный граф

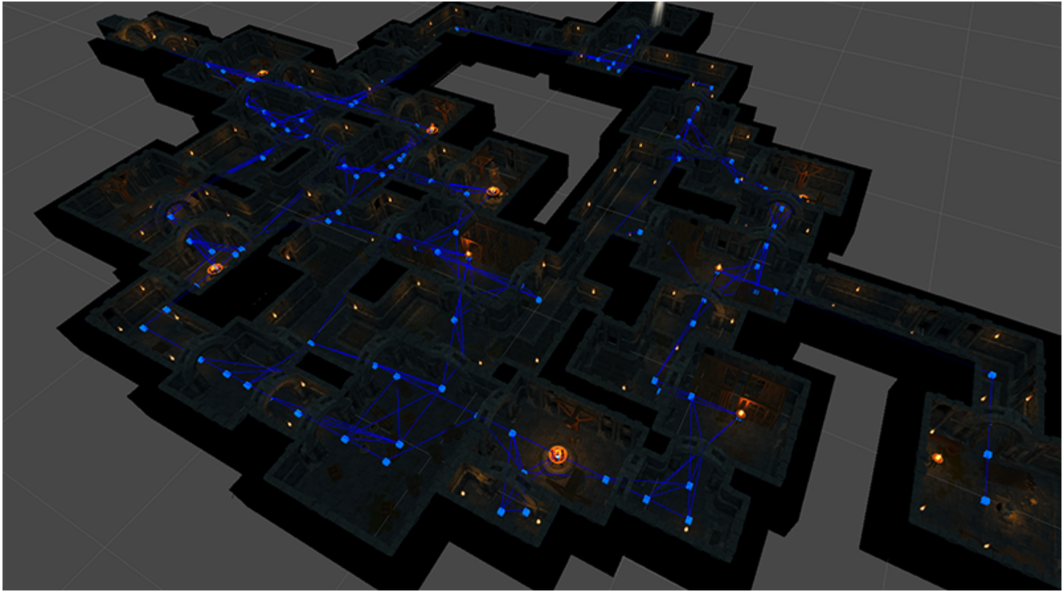
$$G = (V, E)$$

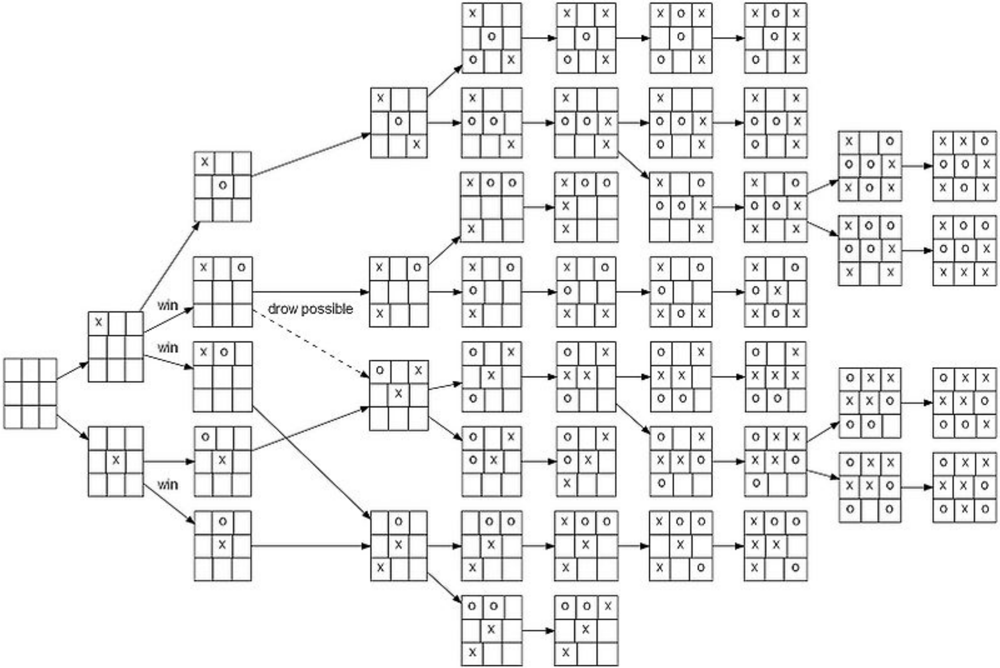
$$E = \{\ \{a, b\}\ :\ a, b \in V\}$$

$$V = \{1, 2, 3, 4, 5\}$$

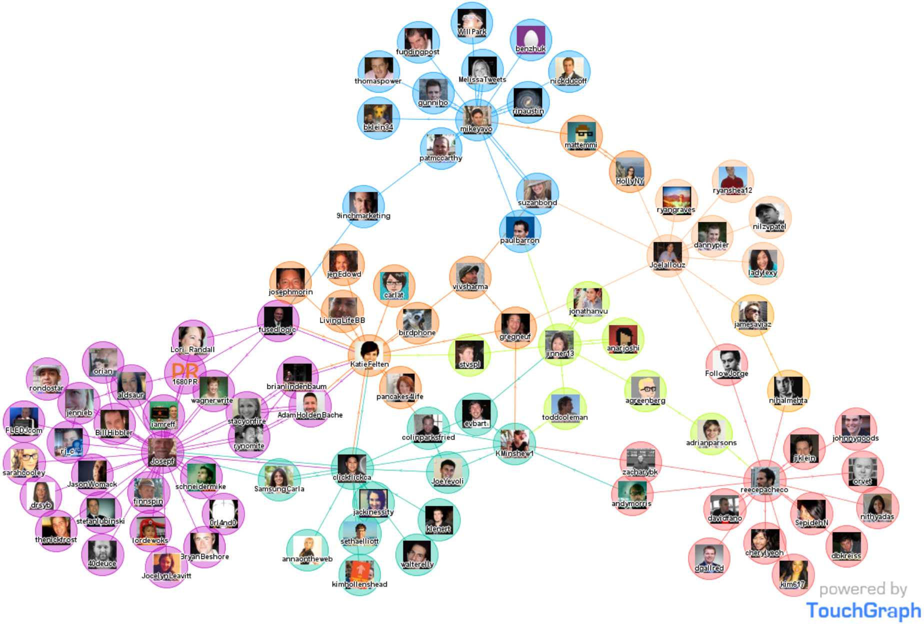
$$E = \{(1,3), (1,5), (3,5), (2,4)\} = \{(a, b)\ :\ a \in V,\ b \in V,\ (a+b)\%2 = 0\}$$







win
win
drow possible
win


Will Park
fundingpost
benzhuk
thomaspower
MelissaTweets
nickducoff
gunniho
rinaustin
bklein34
mikeyavo
mattemmi
patmccarthy
HollyNY
ryanshea12
suzanbond
ryangraves
nilzypatel
9inchmarketing
paulbarron
dannypier
Joelallouz
ladylexy
jenEdowd
josephmorin
carlat
vivsharma
jonathanvu
jamesaviaz
LivingLifeBB
fusedlogic
birdphone
gregneuf
anarjoshi
Lori_Randall
jinner13
KatieFelten
stvspl
FollowJorge
orian
PR
1680PR
rondostar
aldsaur
brianlindenbaum
wagnerwrite
pancakes4life
agreenberg
nihalmehta
jennieb
stacyonfire
AdamHoldenBache
toddcoleman
evbart
FLEDJcom
BillHibbler
iamreff
colinparksfried
adrianparsons
johnnygoods
rj_c
rynomite
KMinshew1
jjklein
sarahcooley
Josepf
zacharybk
orvet
SamsungCarla
clickflickca
JoeYevoli
JasonWomack
schneidermike
reecepacheco
drsyb
andymorris
finnspin
jackinessity
davidfano
SepidehN
nithyadas
stefanlubinski
0rl4nd0
klenert
thenickfrost
lordewoks
sethaelliott
cherylyeoh
dbkreiss
annaontheweb
40deuce
BryanBeshore
walterelly
dgallred
kim617
JocelynLeavitt
kimhollenshead
powered by
TouchGraph

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Степенью вершины в неориентированном графе называют количество инцидентных ей ребер. Для ориентированных графов разделяют степень захода и степень исхода.

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Степенью вершины в неориентированном графе называют количество инцидентных ей ребер. Для ориентированных графов разделяют степень захода и степень исхода.

Путь (маршрут) – это последовательность $v_1, e_1, v_2, e_2, \ldots, v_n$, где $e_i = (v_i, v_j)$, $e_i \in E$, $v_i \in V$.

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Степенью вершины в неориентированном графе называют количество инцидентных ей ребер. Для ориентированных графов разделяют степень захода и степень исхода.

Путь (маршрут) – это последовательность $v_1, e_1, v_2, e_2, \ldots, v_n$, где $e_i = (v_i, v_j)$, $e_i \in E$, $v_i \in V$.

Простой путь – это путь, в котором все ребра и вершины различны.

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Степенью вершины в неориентированном графе называют количество инцидентных ей ребер. Для ориентированных графов разделяют степень захода и степень исхода.

Путь (маршрут) – это последовательность $v_1, e_1, v_2, e_2, \ldots, v_n$, где $e_i = (v_i, v_j)$, $e_i \in E$, $v_i \in V$.

Простой путь – это путь, в котором все ребра и вершины различны.

Цикл (контур) – это путь, в котором первая и последняя вершина совпадают, но других совпадений нет.

Ребро e **инцидентно** вершине v, если $e = (v, w)$ или $e = (w, v)$ для некоторого w

Вершины v и w **индицентны**, если $(v, w) \in E$.

Степенью вершины в неориентированном графе называют количество инцидентных ей ребер. Для ориентированных графов разделяют степень захода и степень исхода.

Путь (маршрут) – это последовательность $v_1, e_1, v_2, e_2, \ldots, v_n$, где $e_i = (v_i, v_j)$, $e_i \in E$, $v_i \in V$.

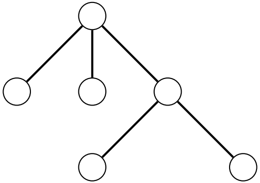
Простой путь – это путь, в котором все ребра и вершины различны.

Цикл (контур) – это путь, в котором первая и последняя вершина совпадают, но других совпадений нет.

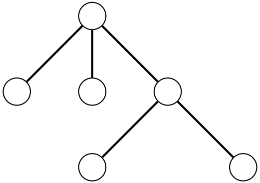
Неориентированный граф **связен**, если между любыми двумя его вершинами существует путь. Для орграфов такое условие называют **сильной связностью**. **Слабая связность** орграфа означает связность соответствующего ему неориентированного графа.

Дерево – это связный неориентированный граф, не содержащий циклов.

Дерево – это связный неориентированный граф, не содержащий циклов.

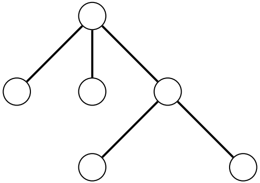


Дерево – это связный неориентированный граф, не содержащий циклов.



Лист – это вершина дерева, имеющая степень 1.

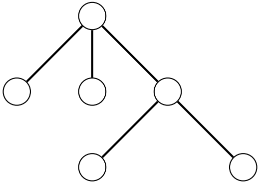
Дерево – это связный неориентированный граф, не содержащий циклов.



Лист – это вершина дерева, имеющая степень 1.

Корень – это произвольно выбранная и зафиксированная вершина дерева.

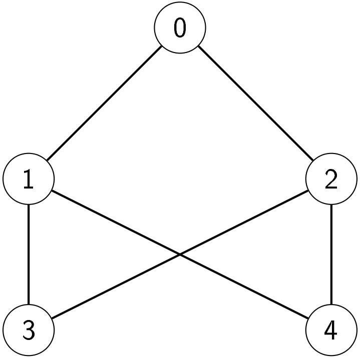
Дерево – это связный неориентированный граф, не содержащий циклов.

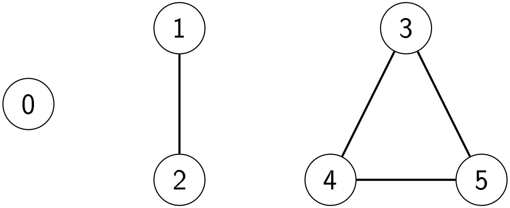


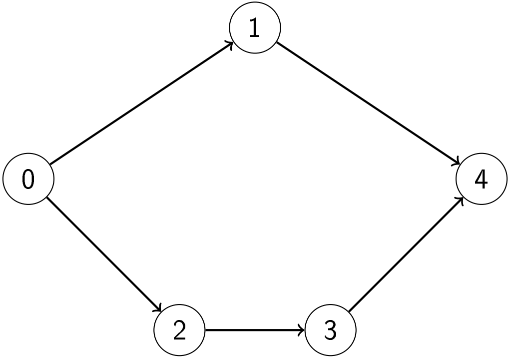
Лист – это вершина дерева, имеющая степень 1.

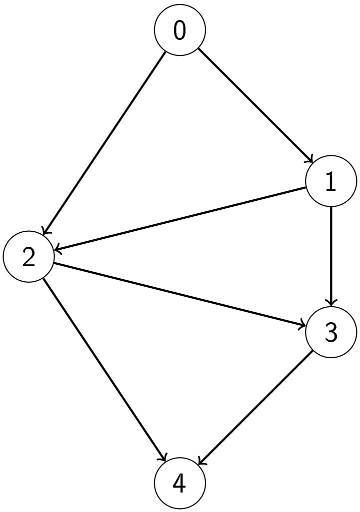
Корень – это произвольно выбранная и зафиксированная вершина дерева.

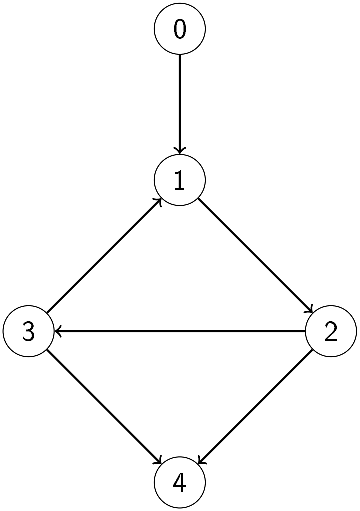
Несвязный неориентированный ациклический граф иногда называют **лесом**.


0
1
2
3
4


0
1
2
3
4
5


0
1
2
3
4


0
1
2
3
4


0
1
3
2
4

Лемма 1. Если в орграфе есть циклы, он не может быть топологически отсортирован.

Лемма 1. Если в орграфе есть циклы, он не может быть топологически отсортирован.

Из любой вершины цикла есть путь в любую другую, а значит, в каком бы порядке мы их не расположили, этот порядок не будет топологической сортировкой

Лемма 1. Если в орграфе есть циклы, он не может быть топологически отсортирован.

Из любой вершины цикла есть путь в любую другую, а значит, в каком бы порядке мы их не расположили, этот порядок не будет топологической сортировкой

Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

Лемма 1. Если в орграфе есть циклы, он не может быть топологически отсортирован.

Из любой вершины цикла есть путь в любую другую, а значит, в каком бы порядке мы их не расположили, этот порядок не будет топологической сортировкой

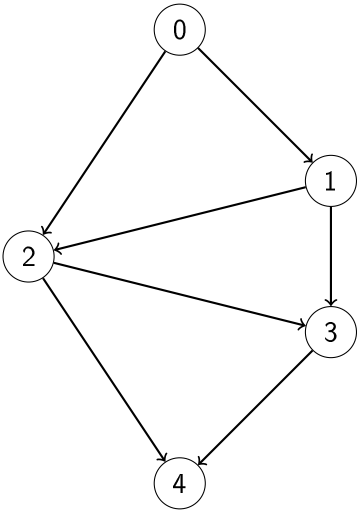
Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

От противного: пусть такой вершины нет, значит, для каждой вершины есть входящее в нее ребро. Возьмем любую v_0, затем v_1 такую, что ребро из v_1 ведет в v_0, и так далее. Эту последовательность можно продолжить бесконечно, но вершин в графе – конечное число, следовательно, вершины повторятся, и образуют цикл.

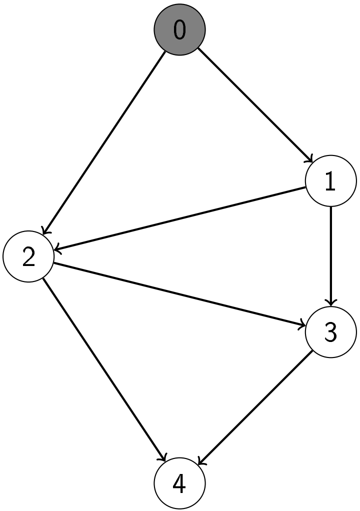
Лемма 3. Если в орграфе нет циклов, он может быть топологически отсортирован

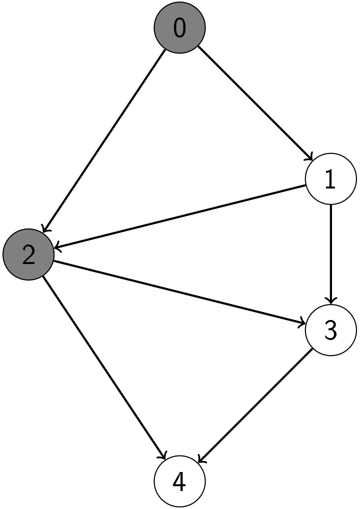
Лемма 3. Если в орграфе нет циклов, он может быть топологически отсортирован

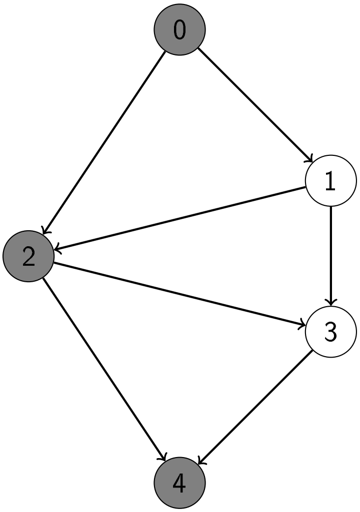
База индукции: очевидно для двухэлементного орграфа.
Шаг индукции. По Лемме 2, есть вершина u с нулевой степенью захода. Исключим вершину u из графа со всеми исходящими ребрами. По предположению индукции, в получившемся графе есть топологическая сортировка $v_1, \ldots, v_n$. Тогда $u, v_1, \ldots, v_n$ будет топологической сортировкой исходного графа: в u нет входящих ребер, а значит, нет и пути из вершин $v_1, \ldots, v_n$.

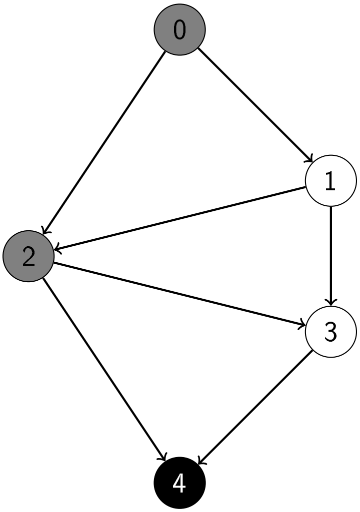


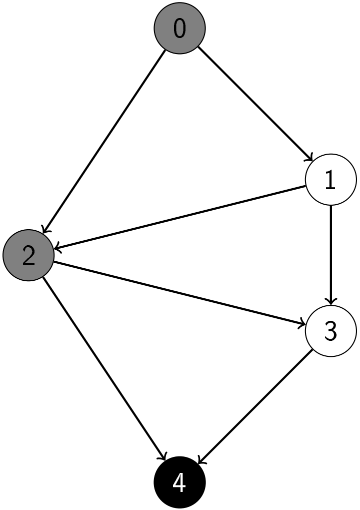
0
1
2
3
4


0
1
2
3
4

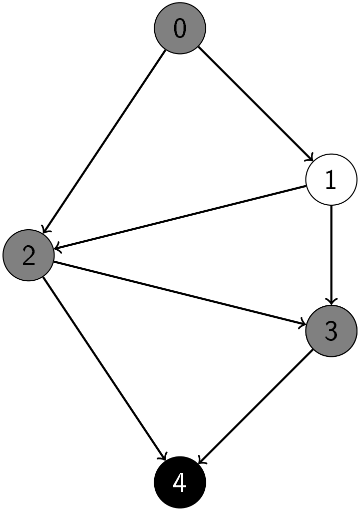

0
1
2
3
4


0
1
2
3
4

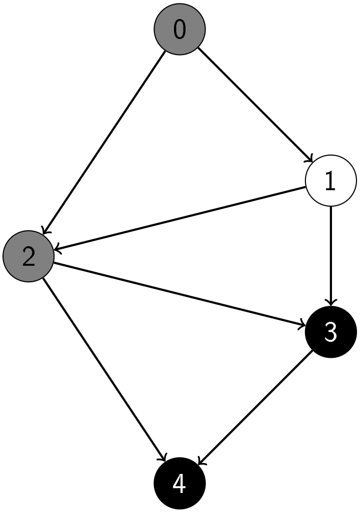

0
1
2
3
4



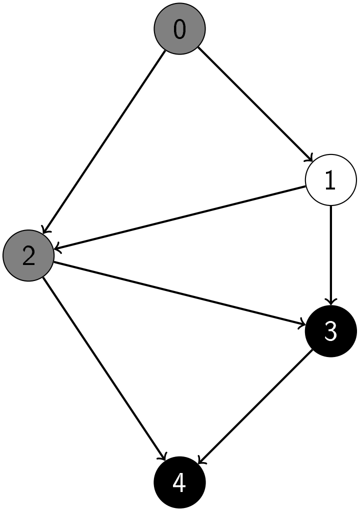
4



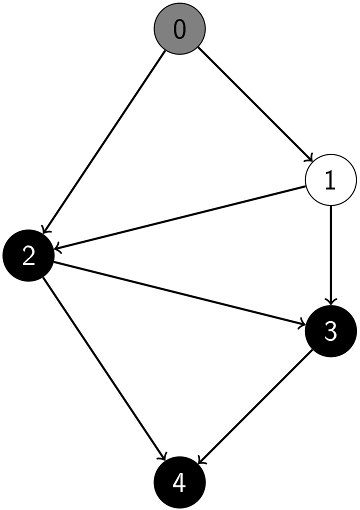
4



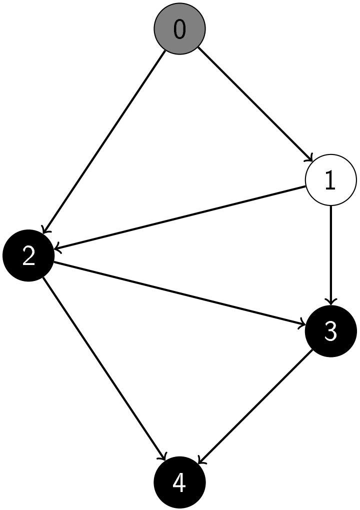
4



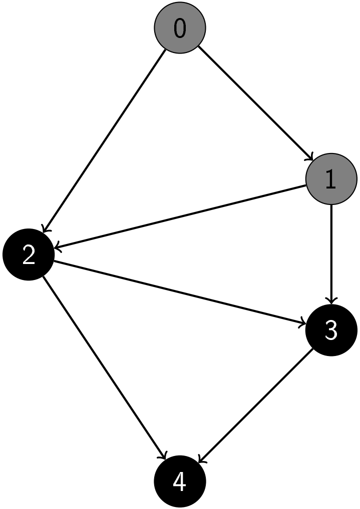
4, 3



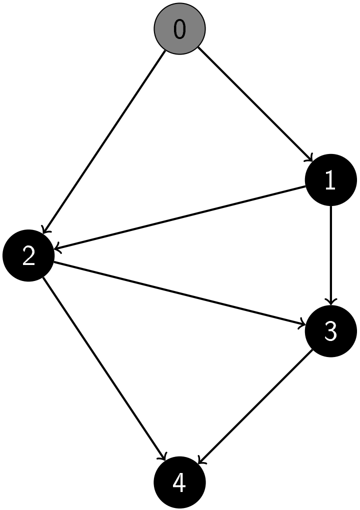
4, 3



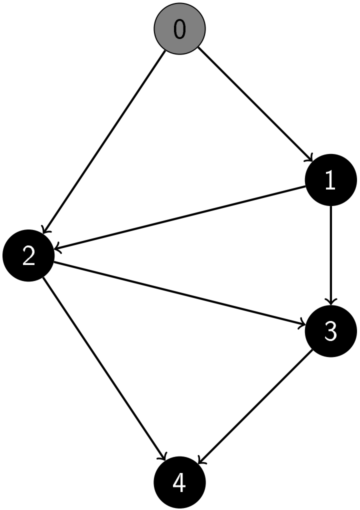
4, 3, 2



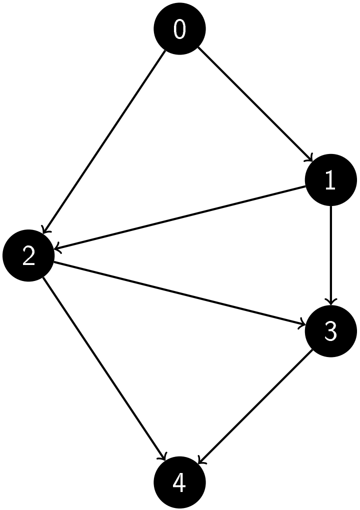
4, 3, 2



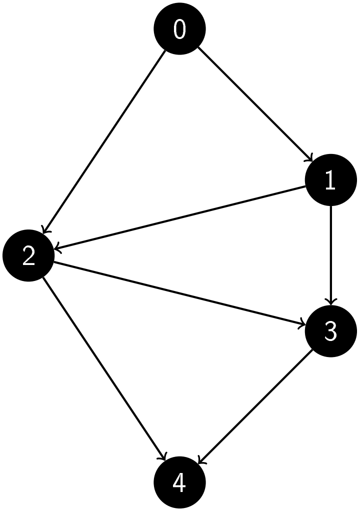
4, 3, 2



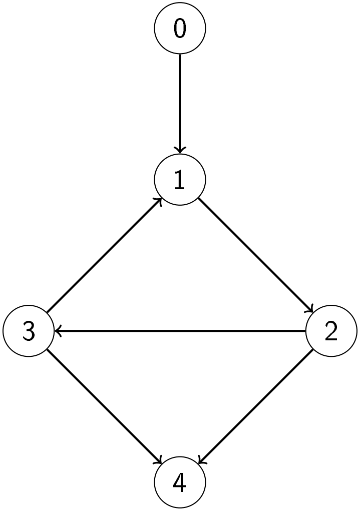
4, 3, 2, 1

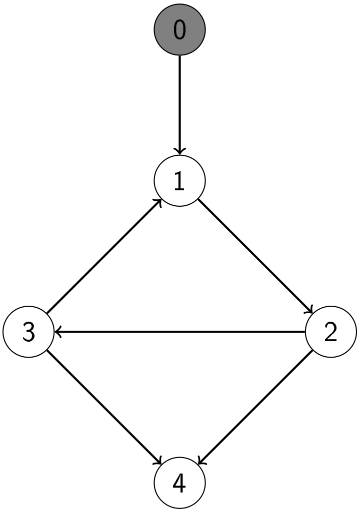


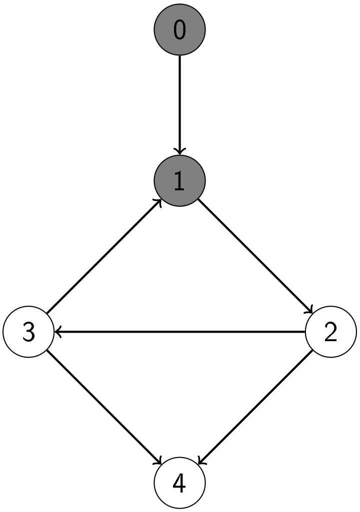
4, 3, 2, 1

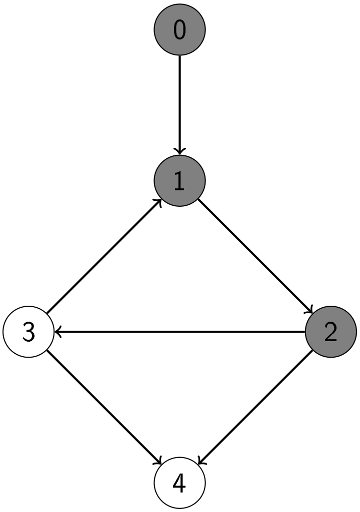


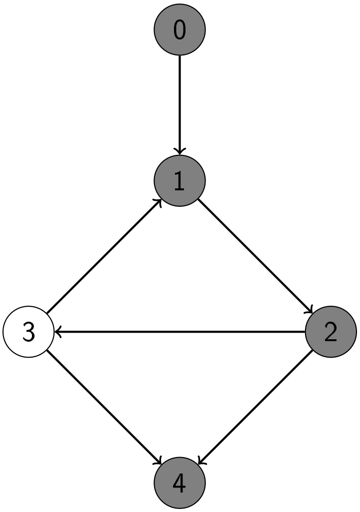
4, 3, 2, 1, 0

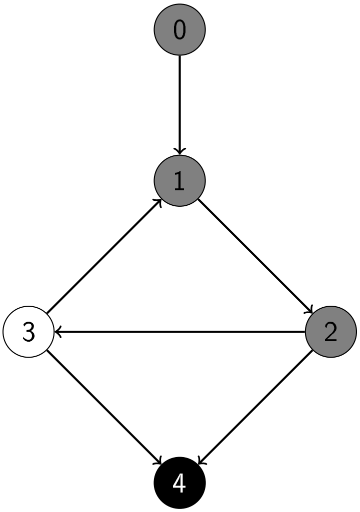

0
1
2
3
4


0
1
2
3
4

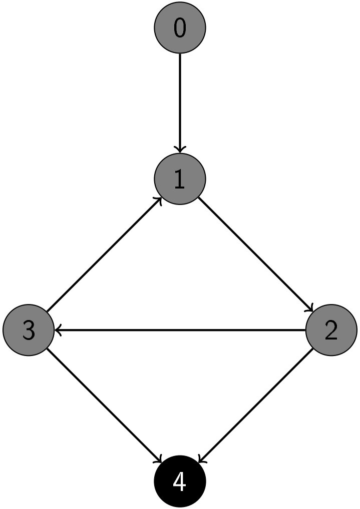

0
1
2
3
4

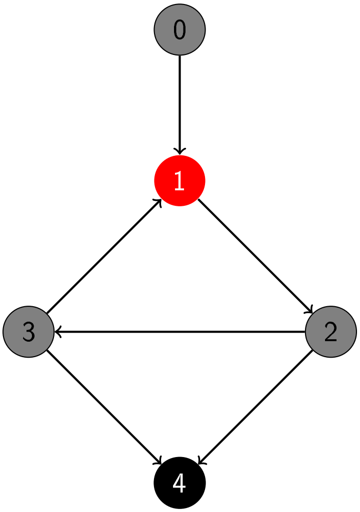

0
1
2
3
4


0
1
2
3
4



0
1
3
2
4


0
1
2
3
4


0
1
2
3
4

Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

Лемма 4. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью исхода.

Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

Лемма 4. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью исхода.

Лемма 5. Если u – вершина с нулевой степенью исхода, а $v_1, \ldots, v_n$ – топологическая сортировка $G \setminus \{u\}$, то $v_1, \ldots, v_n$ – топологическая сортировка G.

Лемма 2. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью захода.

Лемма 4. Если в орграфе нет циклов, то в нем есть вершина с нулевой степенью исхода.

Лемма 5. Если u – вершина с нулевой степенью исхода, а $v_1, \ldots, v_n$ – топологическая сортировка $G \setminus \{u\}$, то $v_1, \ldots, v_n$ – топологическая сортировка G.

Лемма 6. Если в орграфе нет циклов, то алгоритм Тарьяна найдет топологическую сортировку.

Лемма 7. Если в орграфе есть циклы, то алгоритм Тарьяна выдаст ошибку.

Пусть $v_1, \ldots, v_n$ – цикл. Тогда, начав поиск в глубину из одной из вершин v_i, алгоритм отметит эту вершину серым цветом. Затем, пройдя по вершинам $v_{i+1}, v + i + 2, \ldots, v_{i-1}$, он вернется в вершину v_i, которая все еще отмечена серым цветом, и выдаст ошибку.