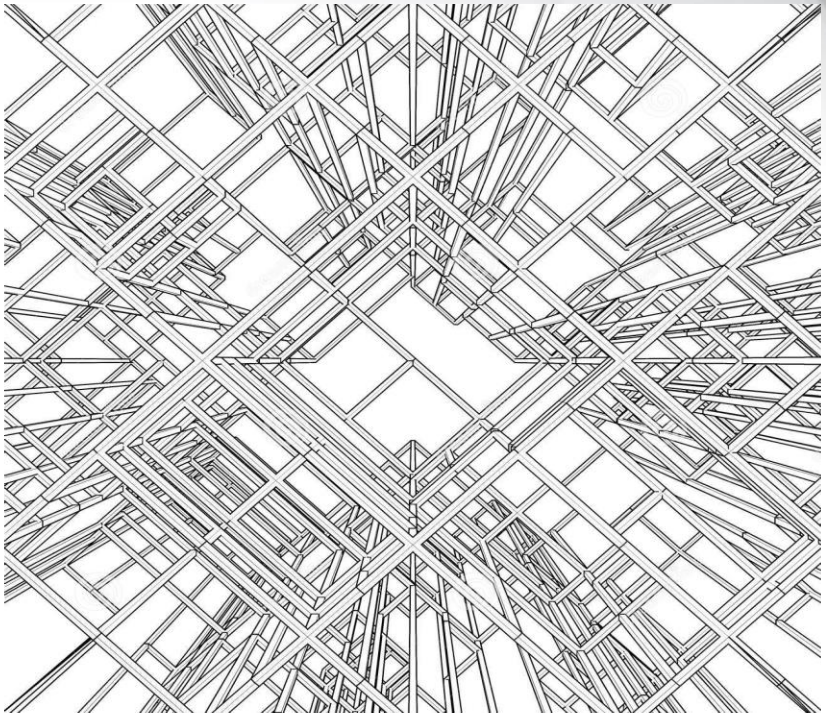
Алгоритмы на графах

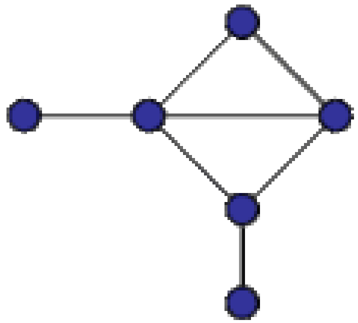


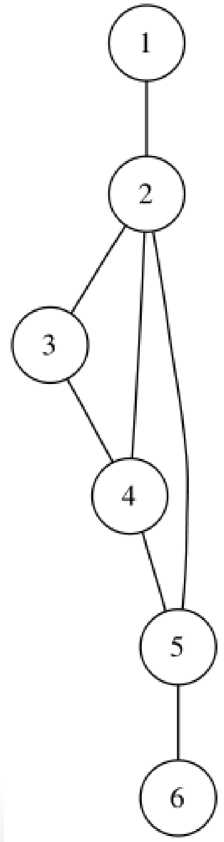
Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

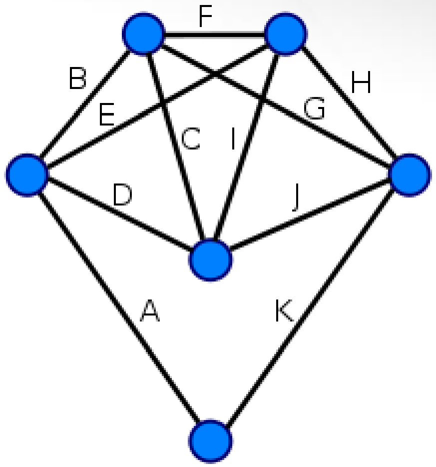
Определения

- **Граф** или **неориентированный граф** *G* — это упорядоченная пара *G*: = (*V*,*E*), для которой выполнены следующие условия:
- *V* это непустое множество **вершин** или **узлов**,
- *E* это множество пар (в случае неориентированного графа — неупорядоченных) вершин, называемых **рёбрами**.

Определения







Определения

- Вершины u и v называются **концевыми** вершинами (или просто **концами**) ребра $e = \{u,v\}$. Ребро, в свою очередь, **соединяет** эти вершины. Две концевые вершины одного и того же ребра называются **соседними**.
- Два ребра называются **смежными**, если они имеют общую концевую вершину.
- Два ребра называются **кратными**, если множества их концевых вершин совпадают.
- Ребро называется **петлёй**, если его концы совпадают, то есть $e = \{v,v\}$.

Определения

- **Ориентированный граф** (сокращённо **орграф**) *G* — это упорядоченная пара *G*: = (*V*,*A*), для которой выполнены следующие условия:
- *V* это непустое множество **вершин** или **узлов**,
- *A* это множество (упорядоченных) пар различных вершин, называемых **дугами** или **ориентированными рёбрами**.

Определение: путь в графе

- **Путём** (или **цепью**) в графе называют конечную последовательность вершин, в которой каждая вершина (кроме последней) соединена со следующей в последовательности вершин ребром.
- **Ориентированным путём** в орграфе называют конечную последовательность вершин v_i (i=1..k), для которой все пары (v_i,v_{i+1}) (i=1..k-1) являются ориентированными рёбрами.

Определение: цикл

Циклом называют путь, в котором первая и последняя вершины совпадают. При этом **длиной** пути (или цикла) называют число составляющих его *рёбер*. Заметим, что если вершины *u* и *v* являются концами некоторого ребра, то согласно данному определению, последовательность (*u*,*v*,*u*) является циклом. Чтобы избежать таких «вырожденных» случаев, вводят следующие понятия.

Определение: цикл

- Всякий путь, соединяющий две вершины, содержит элементарный путь, соединяющий те же две вершины.
- Всякий простой *неэлементарный* путь содержит элементарный *цикл*.
- Всякий *простой* цикл, проходящий через некоторую вершину (или ребро), содержит *элементарный* (под-)цикл, проходящий через ту же вершину (или ребро).
- Петля — элементарный цикл.

Определение: связность

- Бинарное отношение на множестве вершин графа, заданное как «существует путь из *u* в *v*», является отношением эквивалентности, и, следовательно, разбивает это множество на классы эквивалентности, называемые **компонентами связности** графа.
- Если у графа ровно одна компонента связности, то граф связный. На компоненте связности можно ввести понятие **расстояния** между вершинами как минимальную длину пути, соединяющего эти вершины.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Определение: связность

- Всякий максимальный связный подграф графа G называется **связной компонентой** (или просто компонентой) графа G.
  Слово «максимальный» означает максимальный относительно включения, то есть не содержащийся в связном подграфе с большим числом элементов
- Ребро графа называется **мостом**, если его удаление увеличивает число компонент.

Визуализация графов

Для графов с небольшим числом вершин и сопоставимым с ним числом рёбер, самым удобным может быть прямолинейное представление. Примером такой системы может служить дорожная система города. Но для графа социальной сети прямолинейного отображения, из-за большого числа дуг, будет явно недостаточно:

- произвольное;
- прямолинейное — рёбра представляются отрезками;
- сеточное;
- полигональное — для отображения рёбер используются ломаные;
- ортогональное — рёбра представляются ломаными, отрезки которых — вертикальные или горизонтальные линии
- планарное;
- восходящее или нисходящее (для ориентированных графов).

Способы представления графа в информатике

- **Матрица смежности**
  Таблица, где как столбцы, так и строки соответствуют вершинам графа. В каждой ячейке этой матрицы записывается число, определяющее наличие связи от вершины-строки к вершине-столбцу (либо наоборот).

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Способы представления графа в информатике

- **Матрица инцидентности**
  Каждая строка соответствует определённой вершине графа, а столбцы соответствуют связям графа. В ячейку на пересечении *i*-ой строки с *j*-м столбцом матрицы записывается:
  - 1в случае, если связь *j* «выходит» из вершины *i*,
  - −1,если связь «входит» в вершину,
  - 0во всех остальных случаях (то есть если связь является петлёй или связь не инцидентна вершине)

Способы представления графа в информатике

- **Список рёбер** — это тип представления графа в памяти компьютерной программы, подразумевающий, что каждое ребро представляется двумя числами — номерами вершин этого ребра. Список рёбер более удобен для реализации различных алгоритмов на графах по сравнению с матрицей смежности.

Простая задача

В Банановой республике очень много холмов, соединенных мостами. На химическом заводе произошла авария, в результате чего испарилось экспериментальное удобрение «ЗОВАН».

На следующий день выпал цветной дождь, причем он прошел только над холмами. В некоторых местах падали красные капли, в некоторых - синие, а в остальных - зеленые, в результате чего холмы стали соответствующего цвета.

Президенту Банановой республики это понравилось, но ему захотелось покрасить мосты между вершинами холмов так, чтобы мосты были покрашены в цвет холмов, которые они соединяют.

К сожалению, если холмы разного цвета, то покрасить мост таким образом не удастся.

Требуется посчитать количество таких "плохих" мостов.

Простая задача

Входные данные

```
7
0 1 0 0 0 1 1
1 0 1 0 0 0 0
0 1 0 0 1 1 0
0 0 0 0 0 0 0
0 0 1 0 0 1 0
1 0 1 0 1 0 0
1 0 0 0 0 0 0

1 1 1 1 1 3 3
```

Простая задача

Входные данные

7

0 1 0 0 0 1 1

1 0 1 0 0 0 0

0 1 0 0 1 1 0

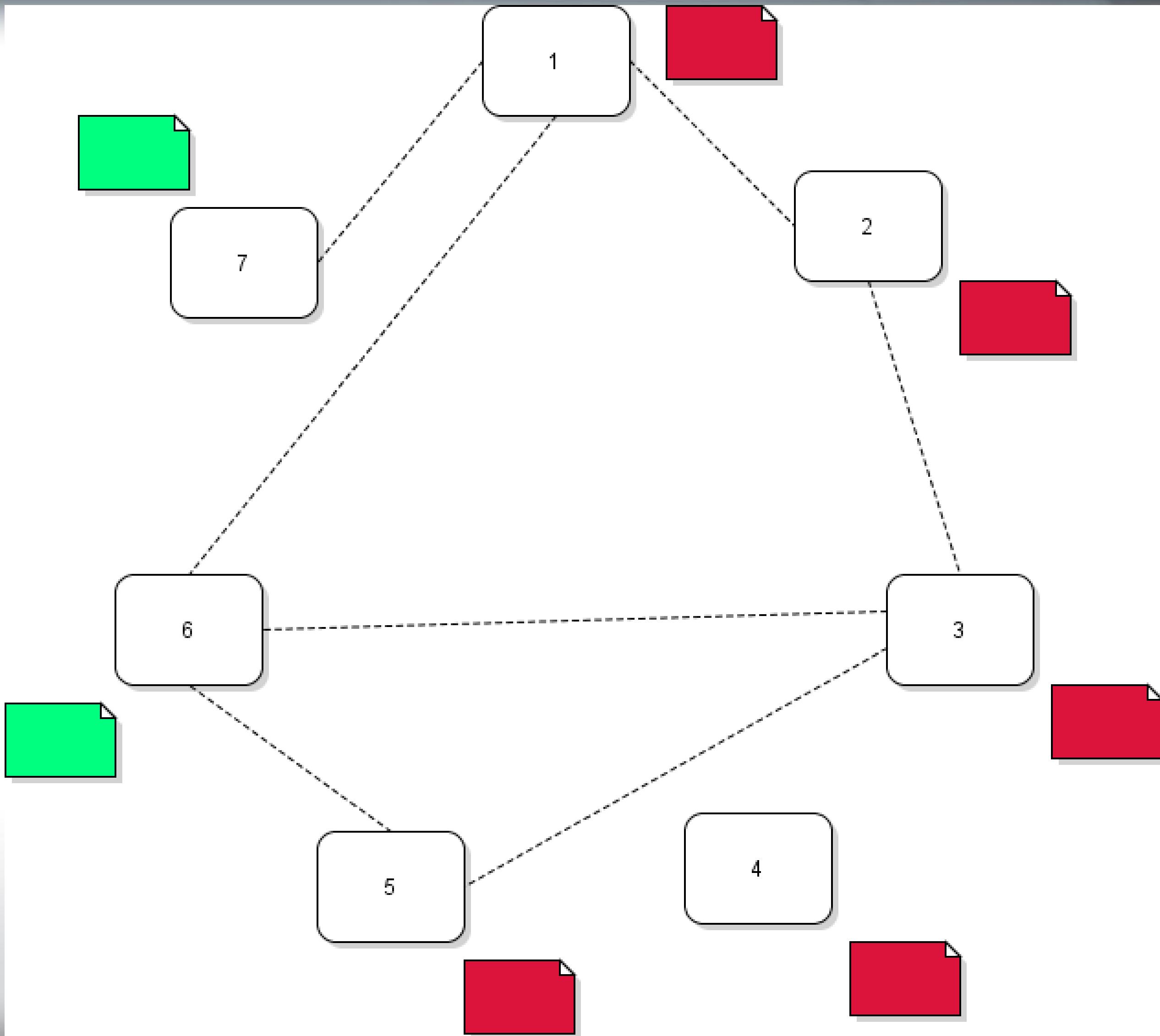
0 0 0 0 0 0 0

0 0 1 0 0 1 0

1 0 1 0 1 0 0

1 0 0 0 0 0 0

1 1 1 1 1 3 3



Топологическая сортировка

Топологическая сортировка — упорядочивание вершин бесконтурного ориентированного графа согласно частичному порядку, заданному ребрами орграфа на множестве его вершин.

Топологическая сортировка

«Генеалогическое дерево» у марсиан

Входные данные

5

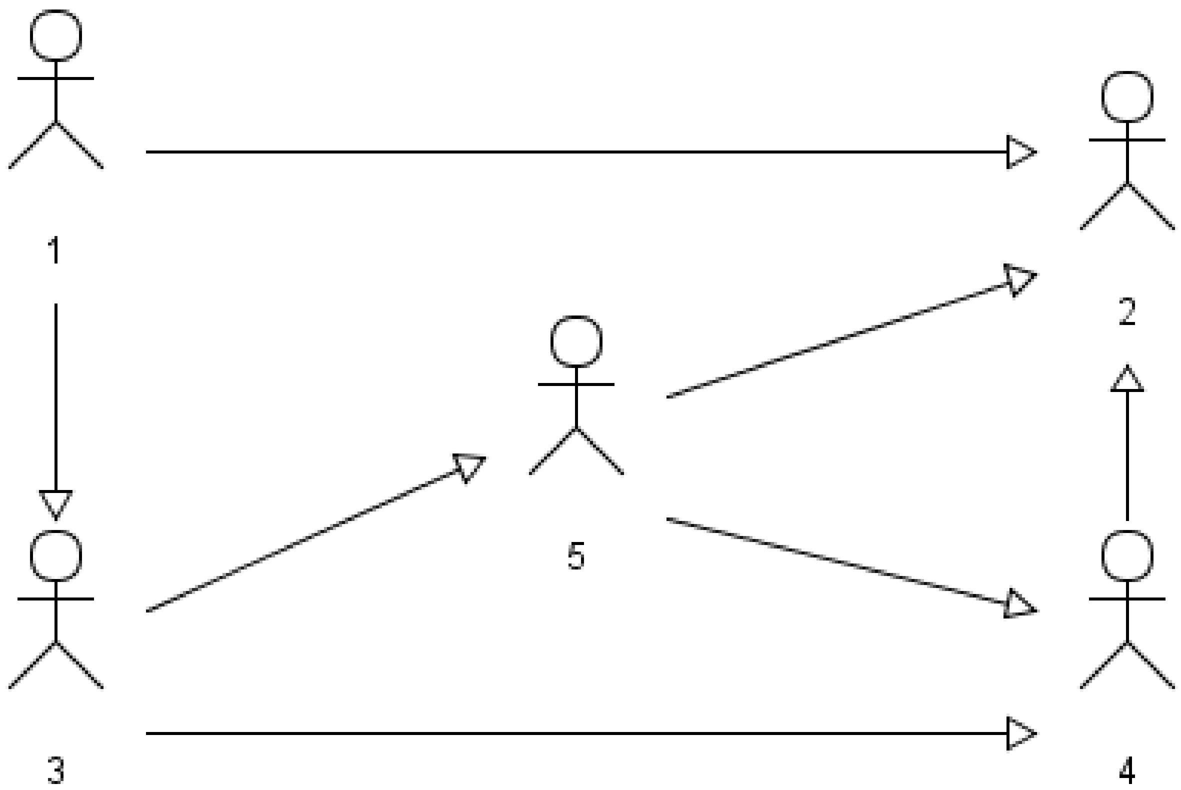
0

4 5 1 0

1 0

5 3 0

3 0



Алгоритм

- Пусть дан бесконтурный ориентированный простой граф $G = (V,E)$.
  Через $A(v), v \in V$ обозначим множество вершин таких, что $u \in A(v) \Leftrightarrow (u,v) \in E$
- То есть, $A(v)$ — множество всех вершин, из которых есть ребро в вершину v.
- Пусть P — искомая последовательность вершин.

Топологическая сортировка - алгоритм

пока | *P* | < | *V* |

выбрать *любую* вершину *v* такую, что $A(v) = \{\varnothing\}$ и $v \notin P$

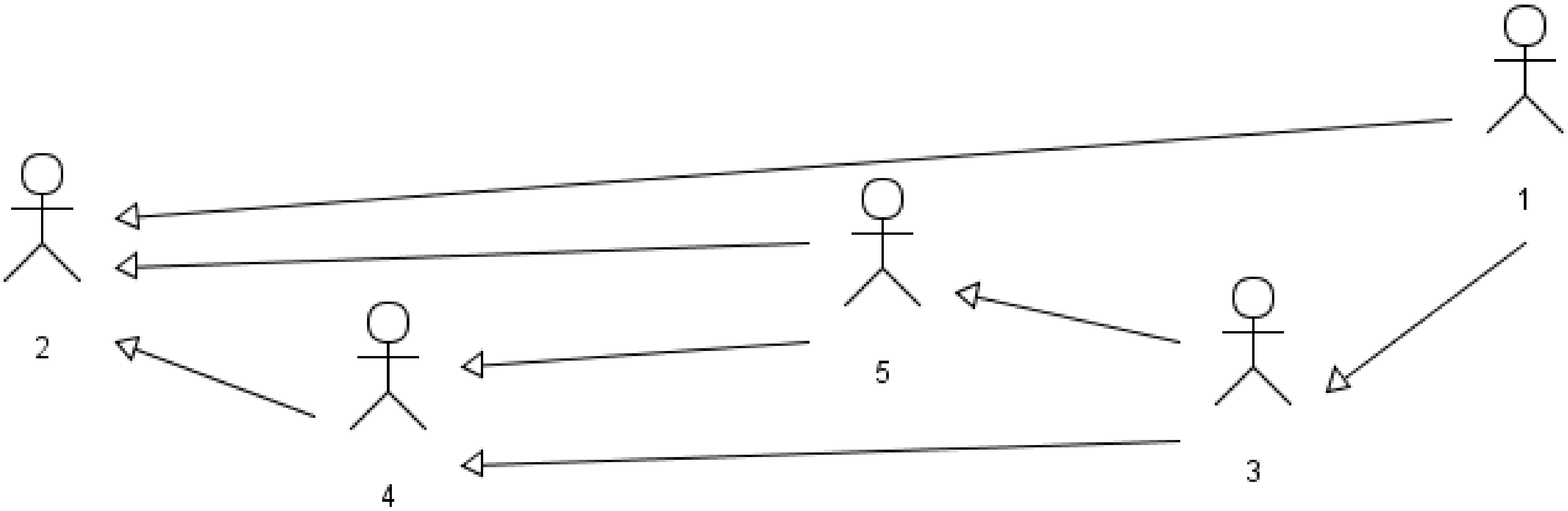
$P \leftarrow P, v$

удалить *v* из всех $A(u), u \neq v$

Наличие хотя бы одного контура в графе приведёт к тому, что на определённой итерации цикла не удастся выбрать новую вершину v.

«Генеалогическое дерево» у марсиан

Результат: 2 4 5 3 1



Поиск в глубину (DFS)

Один из методов обхода графа

Алгоритм поиска описывается следующим образом:

- для каждой не пройденной вершины необходимо найти все не пройденные смежные вершины и
- повторить поиск для них.

Используется в качестве подпрограммы в алгоритмах поиска одно- и двусвязных компонент

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Поиск в ширину (BFS)

Поиск в ширину выполняется в следующем порядке:

- началу обхода s приписывается метка 0, смежным с ней вершинам — метка 1.
- Затем поочередно рассматривается окружение всех вершин с метками 1, и каждой из входящих в эти окружения вершин приписываем метку 2 и т. д.

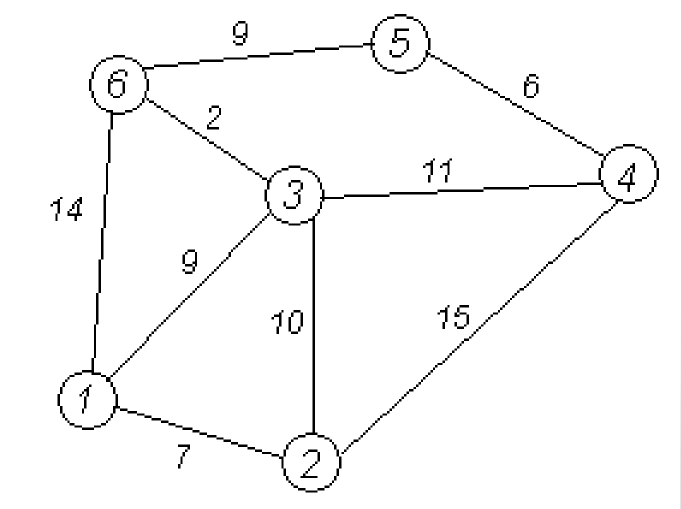
DFS и BFS

- DFS реализуется рекурсивным алгоритмом, либо циклическим алгоритмом со стеком
- BFS может быть получен из циклического алгоритма DFS заменой стека на очередь

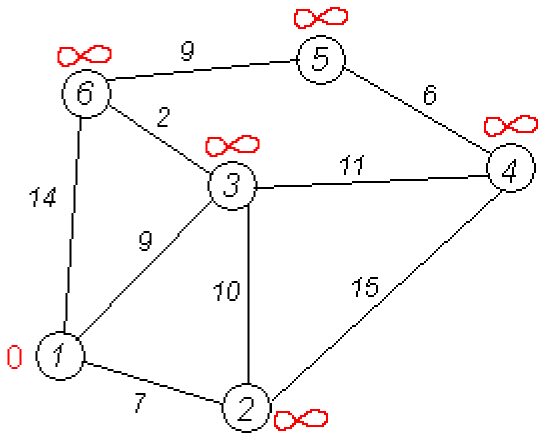
Алгоритм Дейкстры

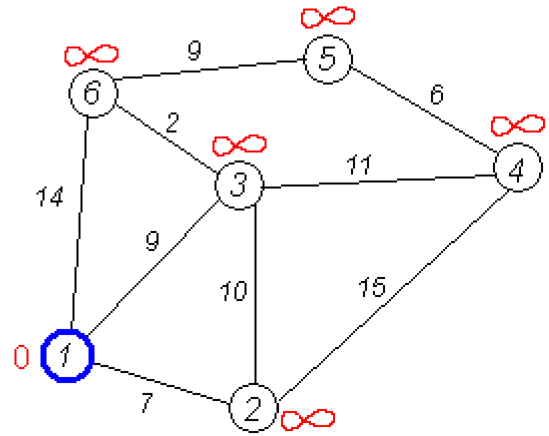
Находит кратчайшее расстояние от одной из вершин графа до всех остальных.

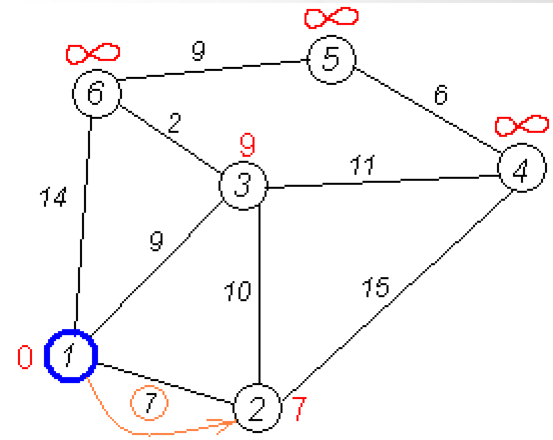
Алгоритм работает только для графов без рёбер отрицательного веса.



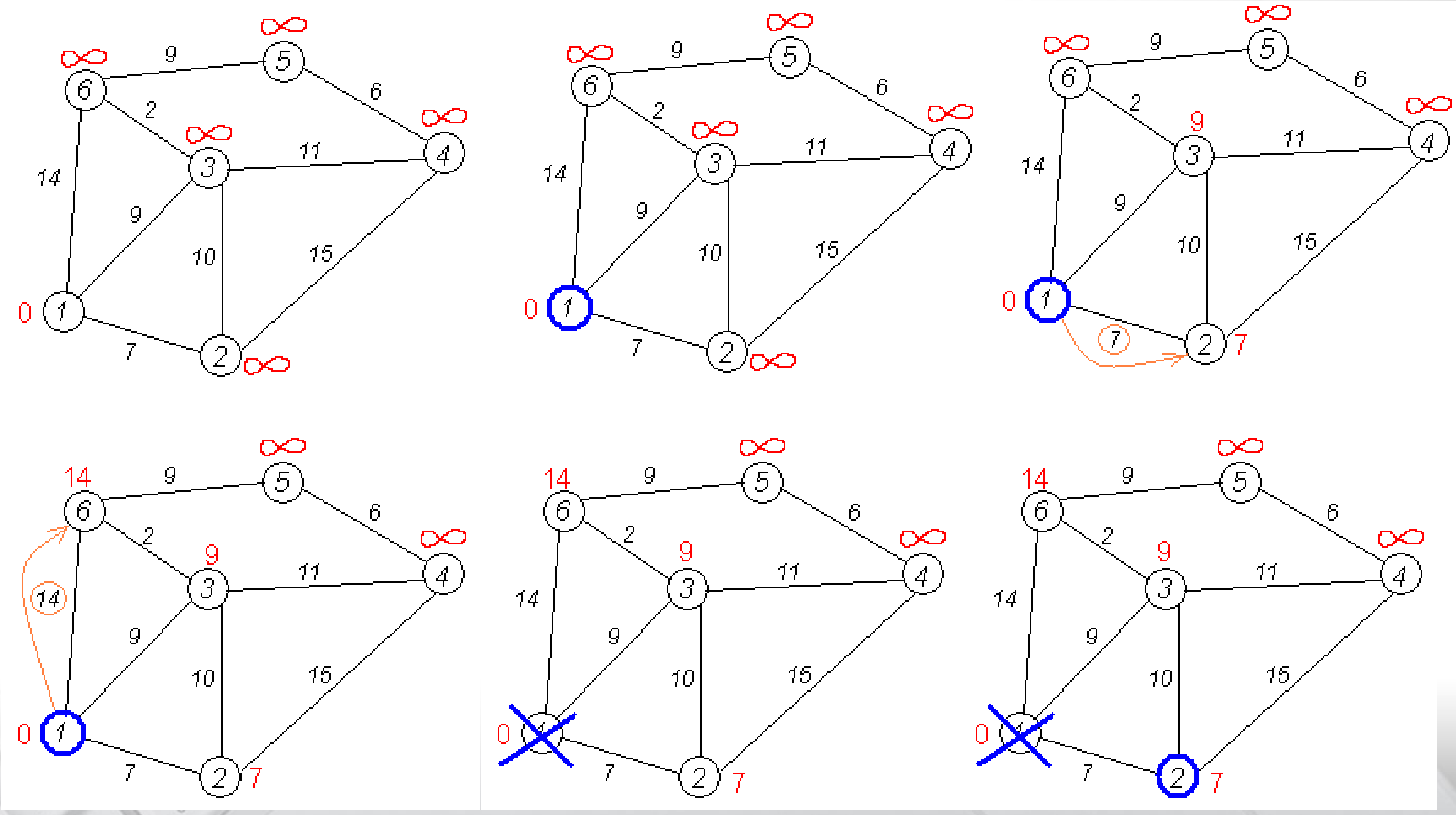
Алгоритм Дейкстры







Алгоритм Дейкстры



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритм Беллмана-Форда

Дан ориентированный или неориентированный граф *G* со взвешенными рёбрами.

Длиной пути назовём сумму весов рёбер, входящих в этот путь.

Требуется найти кратчайшие пути от выделенной вершины *s* до всех вершин графа.

Алгоритм Беллмана-Форда

Для нахождения кратчайших путей от одной вершины до всех остальных, воспользуемся методом динамического программирования.

Построим матрицу A_{ij}, элементы которой будут обозначать следующее: A_{ij} — это длина кратчайшего пути из *s* в *i*, содержащего не более *j* рёбер.

Путь, содержащий 0 рёбер, существует только до вершины *s*. Таким образом, A_{i0} равно 0 при $i = s$, и $+\infty$ в противном случае.

Алгоритм Беллмана-Форда

Теперь рассмотрим все пути из *s* в *i*, содержащие ровно *j* рёбер. Каждый такой путь есть путь из *j* − 1 ребра, к которому добавлено последнее ребро.

Если про пути длины *j* − 1 все данные уже подсчитаны, то определить *j*-й столбец матрицы не составляет труда.

Алгоритм Беллмана-Форда (псевдокод)

for $v \in V$

  do $d[v] \leftarrow +\infty$

$d[s] \leftarrow 0$

for $i \leftarrow 1$ to $|V| - 1$

  do for $(u, v) \in E$

    if $d[v] > d[u] + w(u,v)$

      then $d[v] \leftarrow d[u] + w(u, v)$

return d

Алгоритм Беллмана-Форда

- Вместо массива *d* можно хранить всю матрицу А, но это требует $O(V^2)$ памяти. Зато при этом можно вычислить и сами кратчайшие пути, а не только их длины. Для этого заведем матрицу P_{ij}.
- Если элемент A_{ij} содержит длину кратчайшего пути из *s* в *i*, содержащего *j* рёбер, то P_{ij} содержит предыдущую вершину до *i* в одном из таких кратчайших путей (ведь их может быть несколько).

Алгоритм Беллмана-Форда

for $v \in V$
  for $i \leftarrow 0$ to $|V| - 1$
    do $A_{vi} \leftarrow +\infty$
$A_{s0} \leftarrow 0$
for $i \leftarrow 1$ to $|V| - 1$
  do for $(u, v) \in E$
    if $A_{vi} > A_{u,i-1} + w(u,v)$
      then $A_{vi} \leftarrow A_{u,i-1} + w(u, v)$
        $P_{vi} \leftarrow u$

Алгоритм Флойда–Уоршелла

Динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа.

Алгоритм Флойда–Уоршелла

- На каждом шаге алгоритм генерирует двумерную матрицу *W.*
- Матрица *W* содержит длины кратчайших путей между всеми вершинами графа.
- Перед работой алгоритма матрица *W* заполняется длинами рёбер графа.

Алгоритм Флойда-Уоршелла

Псевдокод:

```
for k = 1 to n
 for i = 1 to n
  for j = 1 to n
    W[i][j] =
      min(W[i][j], W[i][k]+W[k][j])
```

Задачи

Простая задача	Цветной дождь: http://www.e-olymp.com/ru/problems/994
Топологическая сортировка	Генеалогическое дерево: http://www.e-olymp.com/ru/problems/2696
Поиск в глубину	Площадь комнаты: http://www.e-olymp.com/ru/problems/4001 Покраска лабиринта: http://www.e-olymp.com/ru/problems/1061 Удаление клеток: http://www.e-olymp.com/ru/problems/1063
Поиск в ширину	Один конь: http://www.e-olymp.com/ru/problems/997 Путь коня: http://www.e-olymp.com/ru/problems/1064 Линии: http://www.e-olymp.com/ru/problems/1060

Еще задачи

Уборка снега:
http://www.e-olymp.com/ru/problems/61
Города и дороги:
http://www.e-olymp.com/ru/problems/992
Алгоритм Дейкстры:
http://www.e-olymp.com/ru/problems/1365
Флойд:
http://www.e-olymp.com/ru/problems/975
Заправки:
http://www.e-olymp.com/ru/problems/1388
Шайтан-машинка:
http://www.e-olymp.com/ru/problems/4850

НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА