Модуль 3. NP-трудные задачи

Лекция 12

Задача изоморфизма графов и подграфов.

Задача коммивояжёра.

Что делать?

1) Решать долго (за экспоненциальное время)
2) Пытаться сократить время решения для входов, ожидаемых на практике
   – Полиномиальные в среднем алгоритмы
   – Параметризованные алгоритмы
3) Решать приближённо
   – С гарантированной *оценкой точности*
   – Без гарантий, но как правило достаточно точно
   – С некоторой *вероятностью* ошибки

Метод полного перебора

Полный перебор (Brute Force)

- Последовательно генерировать все возможные решения
- Для каждого сгенерированного решения x выполнять проверку на допустимость / оптимальность: *Process*(x)

Генерация перестановок

Задача: для заданного n сгенерировать и обработать все перестановки степени n.

Решение:

1. Храним в массиве $A[1..n]$.
2. Инициализация: $\forall\, i \quad A[i] := i$.
3. Для всех k последовательно переставляем $A[k]$ с элементами в позициях $1,\ldots, k-1$.

Генерация перестановок

Вызов: ProcessPermutations(*A*,*n*)

<u>ProcessPermutations(*A*,*k*)</u>

if *k* = 1 then Process(*A*)

else

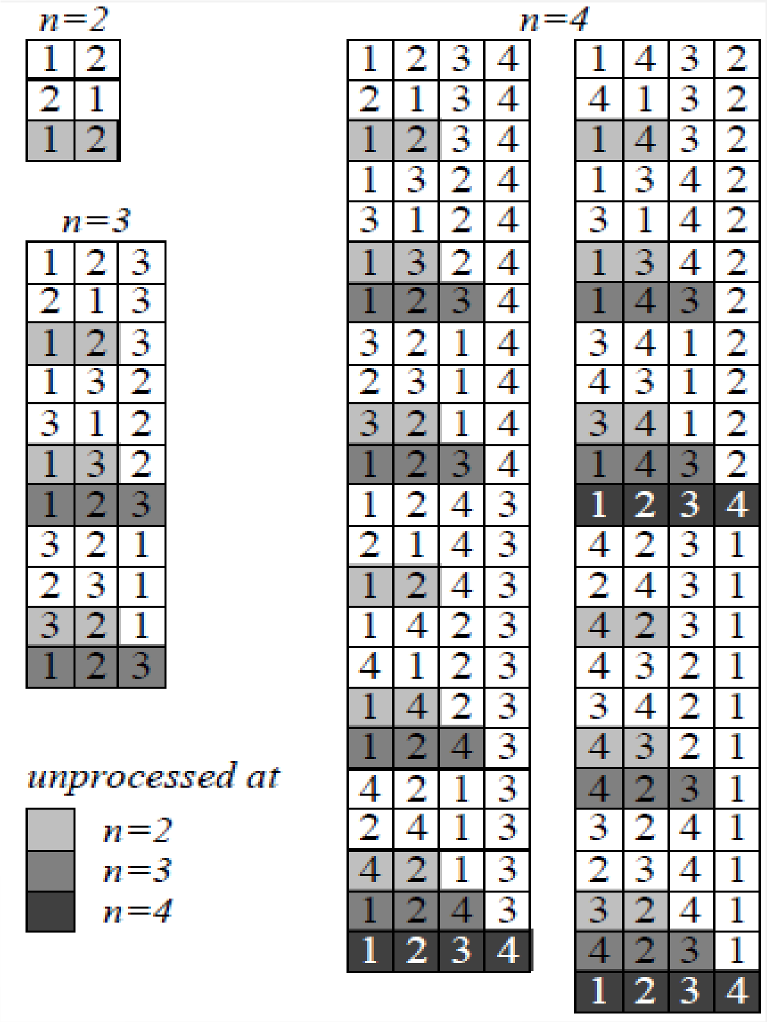
    ProcessPermutations(*A*, *k*-1);

    for *i* = *k*-1 downto 1 do

        Поменять A[*k*] и A[*i*]

        ProcessPermutations(*A*, *k*-1);

        Поменять A[*k*] и A[*i*]

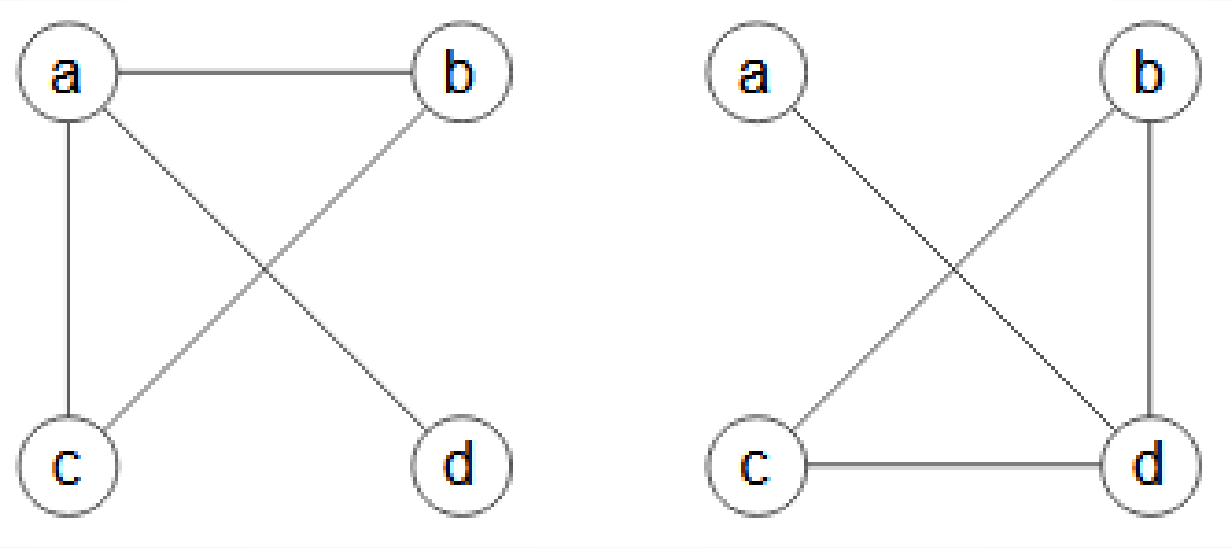


Изоморфизм графов

Изоморфизм графов

Определение. Графы $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ *равны*, если их множества вершин и множества рёбер совпадают: $V_1 = V_2, E_1 = E_2$.

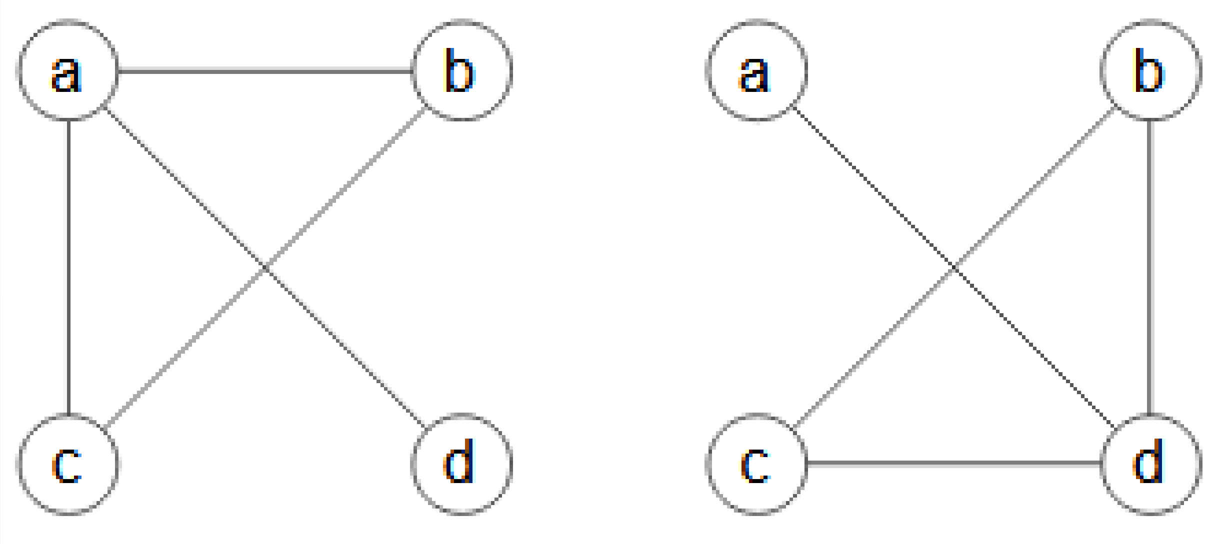
Равенство графов можно определить по матрице смежности.



Изоморфизм графов

Более «мягкое» понятие сходства графов отражается отношением *изоморфизма* графов.

Определение. Графы $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ *изоморфны*, если существует взаимно однозначное отображение $\varphi: V_1 \to V_2$, такое, что $(u, v) \in E_1 \Leftrightarrow \big(\varphi(u), \varphi(v)\big) \in E_2$.



Изоморфизм графов

Задача изоморфизма графов:

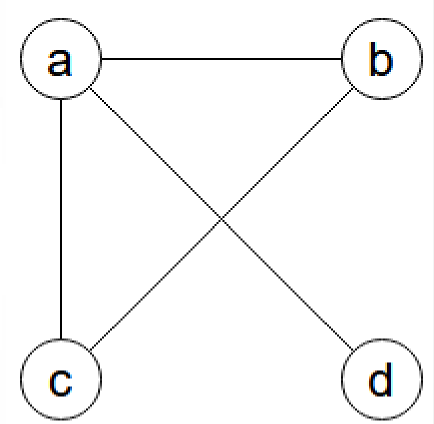
- Дано: два графа $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$.
- Вернуть: изоморфны ли графы G_1 и G_2?

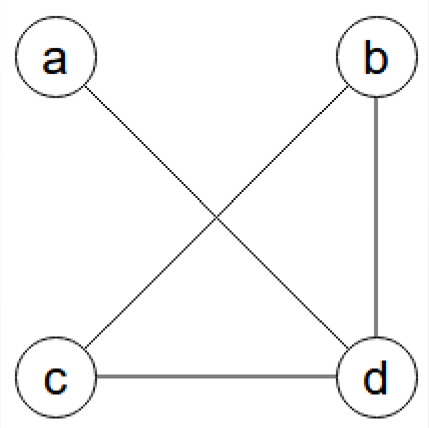
То есть, это распознавательная задача.

Стандартное обозначение изоморфизма графов:
$G_1 \cong G_2$.

Изоморфизм графов

Взаимно однозначное отображение конечного множества – это аналог перестановки. Поэтому проверить изоморфизм можно полным перебором всех перестановок из $n(= |V_1|)$ элементов.





v	$\varphi(v)$
a	d
b	b
c	c
d	a

Изоморфизм графов

Сложность переборного алгоритма: $O(n!)$.
По формуле Стирлинга

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

получаем временную сложность переборного алгоритма: $O(n^n)$.

На текущий момент, для задачи проверки изоморфизма графов не построен полиномиальный алгоритм, но и не доказана её NP-полнота.

Изоморфизм подграфов

Задача изоморфизма *подграфов*:

- Дано: два графа $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$, $|V_1| > |V_2|$.
- Вернуть:
  1) Распознавательный вариант: содержит ли G_1 подграф H, изоморфный ли графу G_2 $(H \cong G_2)$?
  2) Вычислительный вариант: найти в G_1 подграф H, изоморфный ли графу G_2 $(H \cong G_2)$, или сообщить, что такого подграфа нет.

Задача изоморфизма подграфов является NP-трудной.

Задача коммивояжёра

Определения

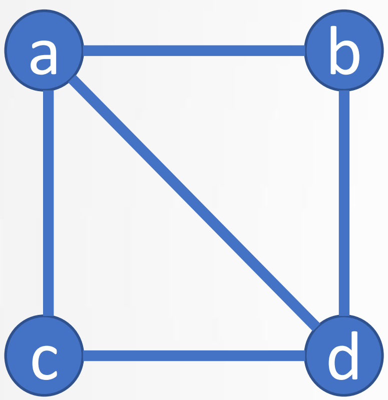
Рассмотрим $G(V,E)$ – связный граф.

Определения

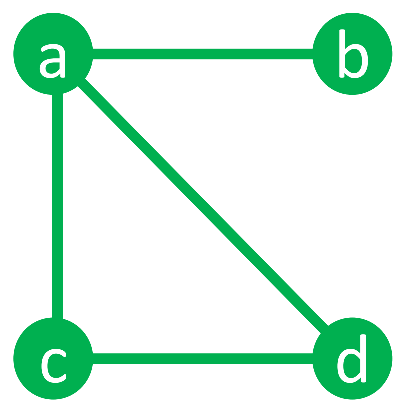
Цикл Z (путь P) называется ***гамильтоновым циклом (путём)*** на G, если и только если Z (P) проходит ровно один раз через каждую вершину графа G.

Граф $G(V,E)$ называется *гамильтоновым* (*полугамильтоновым*), если на нём существует гамильтонов цикл (путь).

Определения

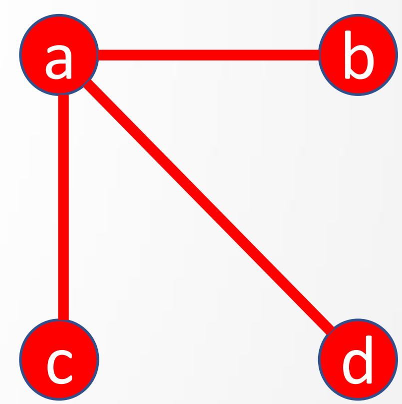


Гамильтонов граф



Полугамильтонов граф

Негамильтонов граф



Определения

Пусть $G(V, E)$ – связный граф; $w: E \longrightarrow R_+$ - весовая функция.

Вес цикла Z (пути P) определим как $w(Z) = \sum_{e \in Z} w(e)$.

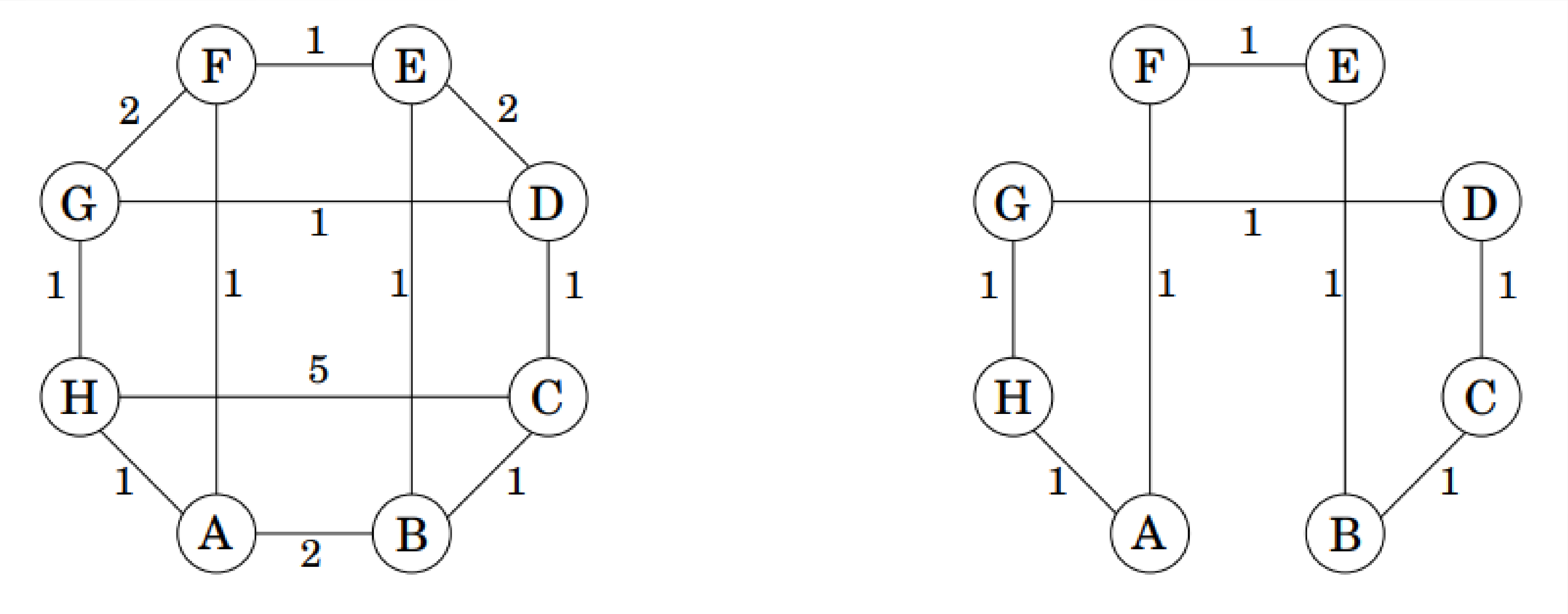
Распознавательная задача: является ли граф $G(V, E)$ (полу)гамильтоновым?

Вычислительная задача: построить на $G(V, E)$ гамильтонов цикл (вернуть ‘NULL’, если $G(V, E)$ не гамильтонов).

Оптимизационная задача (=задача коммивояжёра, Travelling Salesman Problem, TSP): построить минимальный гамильтонов цикл на графе $G(V, E)$ (вернуть ‘NULL’, если $G(V, E)$ не гамильтонов).

Определения

Пример графа и оптимального гамильтонова цикла на нём:



http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf

Решение ЗК

<u>Теорема 1</u>: Задача коммивояжёра является NP-трудной.

Что делать?

1) Решать долго (за экспоненциальное время)

– Пытаться сократить перебор за счёт доп. анализа (*метод ветвей и границ*)

1) Пытаться сократить время решения для входов, ожидаемых на практике

– Полиномиальные в среднем алгоритмы

– Параметризованные алгоритмы

3) Решать приближённо

– С гарантированной *оценкой точности*

– Без гарантий, но как правило достаточно точно (*эвристики*)

– С некоторой *вероятностью* ошибки

Решение ЗК

<u>**Определение**</u>: Задача коммивояжёра называется *метрической* (MTSP), если весовая функция $w: E \longrightarrow R_+$ является метрикой, то есть удовлетворяет аксиомам метрики:

1) $w(u,v) \geq 0$; $w(u,v)=0 \Leftrightarrow u=v$

2) $w(u,v) = w(v,u)$

3) неравенство треугольника: $w(u,v) \leq w(u,x) + w(x,v)$

Решение ЗК

Метрическая ЗК является важным частным случаем общей ЗК.

Важным частным случаем метрической ЗК является евклидова ЗК: вершины графа представляют точки в R^n, а вес ребра = евклидову расстоянию между вершинами.

Решение ЗК

Теорема 2: Метрическая задача коммивояжёра является NP-трудной.

Теорема 3: Даже евклидова задача коммивояжёра является NP-трудной.

Решение ЗК

Рассмотрим переборное (=точное, но медленное) решение общей ЗК.

Решение задачи (гамильтонов цикл или путь) можно представить минимум двумя способами:

1) Перестановка вершин: список вершин в том порядке, в котором через них проходит цикл/путь.
2) Последовательность рёбер: список рёбер в том порядке, в котором через них проходит цикл/путь.

Представление в виде перестановки вершин немного удобрее, т.к. для проверки корректности достаточно проверить, что соседние в перестановке вершины смежны на графе (включая первую и последнюю вершины в случае замкнутой задачи коммивояжёра). При использовании последовательности рёбер проверка корректности выполняется сложнее.

Решение 3К

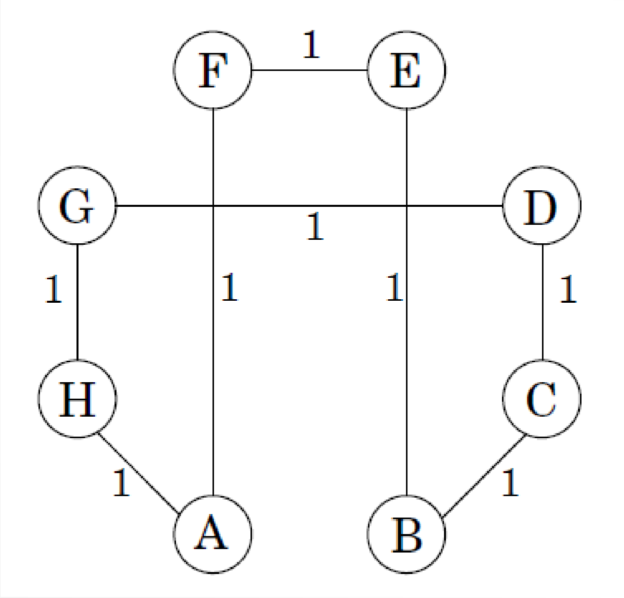
Алгоритм решения с помощью перебора ***перестановок***:

- Пусть $n = |V|$.
- Контур можно задать как перестановку номеров вершин.
- Но считаем, что в конце всегда стоит v_n.
- Поэтому перебираем только перестановки чисел $\{1, \ldots, n\text{-}1\}$.

Временная сложность: $O((n\text{-}1)!)$.

Решение ЗК

Для поиска оптимального *цикла* нужно сгенерировать $(n-1)!$ перестановок.



AFEBCDGH

FEBCDGHA

EBCDGHAF

…

Для неориентированного графа: только $\frac{(n-1)!}{2}$:

AHGDCBEF

HGDCBEFA

…

Генерация перестановок

Вызов: ProcessPermutations(*A*,*n*)

<u>ProcessPermutations(*A*,*k*)</u>

if *k* = 1 then Process(*A*)

else

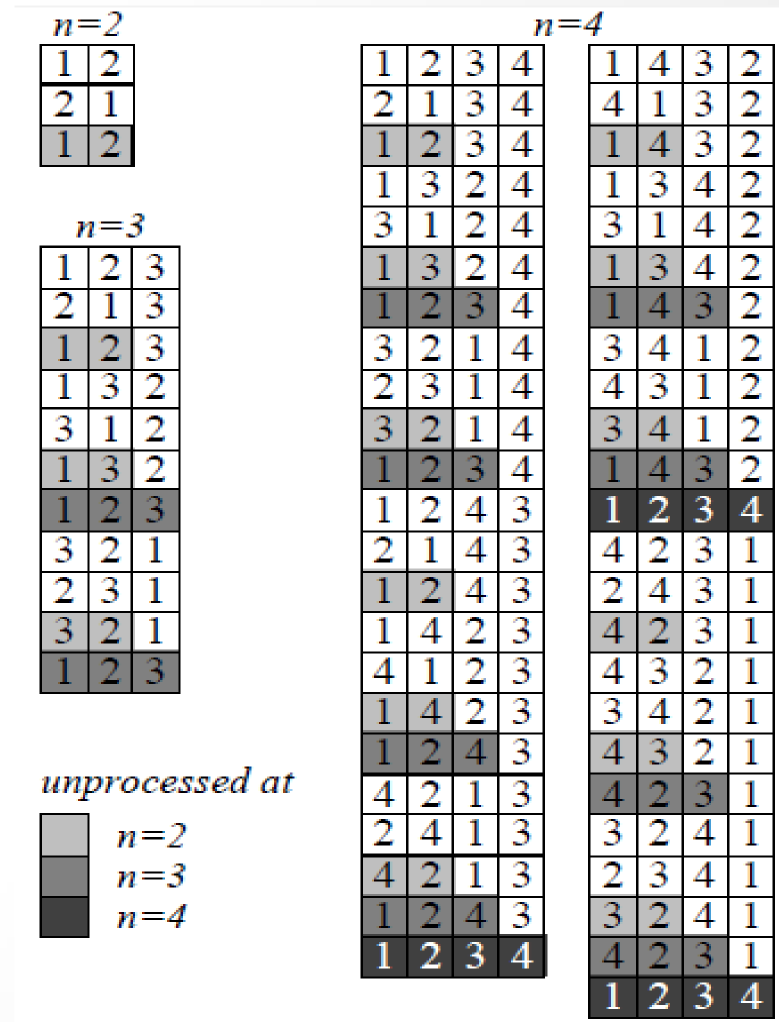
    ProcessPermutations(*A*, *k*-1);

    for *i* = *k*-1 downto 1 do

        Поменять A[*k*] и A[*i*]

        ProcessPermutations(*A*, *k*-1);

        Поменять A[*k*] и A[*i*]

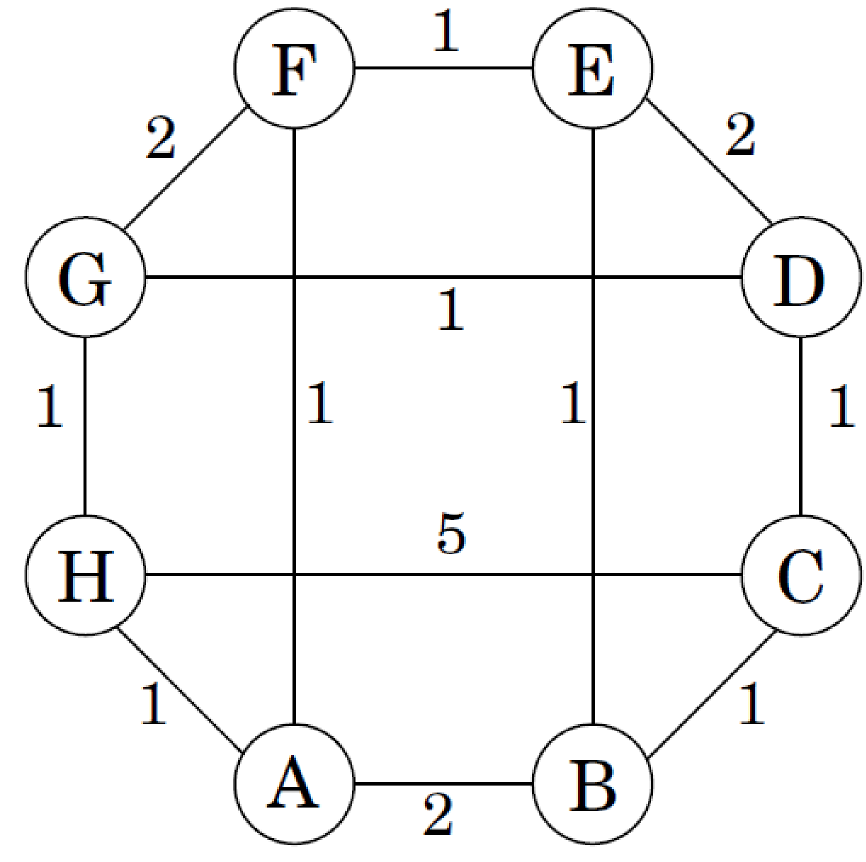


Решение ЗК

Для чего нужна процедура `Process()`?

1. Надо проверить корректность – представляет ли текущая перестановка допустимый (гамильтонов) цикл на заданном графе.
2. Если перестановка представляет корректный цикл, посчитать его вес и сравнить с весом цикла-*рекордсмена*.

Решение ЗК



Пример:

- Сгенерировать 7! перестановок, зафиксировав вершину A в качестве начальной.
- Перестановка ‘aBCDEFGH’ допустима, её вес = 11.
- Перестановка ‘aBCDEFHG’ не допустима, т.к. вершины F и H не смежны на графе.
- Перестановка ‘aFEBCHGD’ не допустима (не представляет гамильтонов цикл), т.к. вершина D не смежна с вершиной A.