Институт проблем передачи информации РАН
Лаборатория «Математических
методов и моделей в биоинформатике»
http://lab6.iitp.ru/ru/pub/
Мех-мат МГУ
«Кафедра математической логики и
теории алгоритмов»
http://lpcs.math.msu.su/rus/staff.htm
адрес: **lyubetsk@iitp.ru**

В. Любецкий, К. Горбунов

Оптимальное преобразование графов и линейное программирование, предельные вычисления

Теория множеств была и остаётся основой основ, но в своё время произошёл перенос центра тяжести на физические и инженерные приложения, где нет дискретности (всё непрерывно).

Графы – теория любых конечных множеств и конечных отношений.

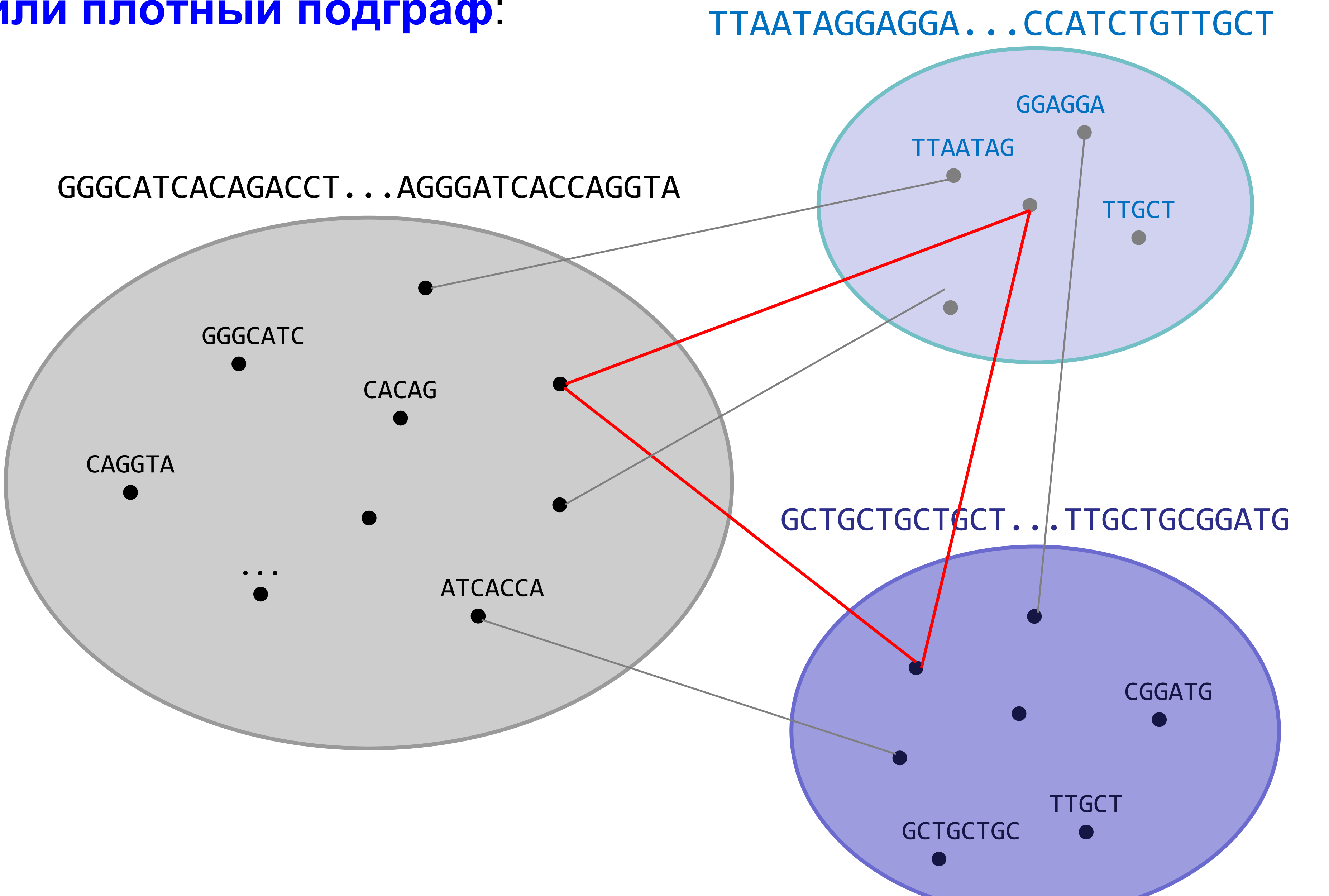
Второе (очень близкое) ключевое понятие – **Большие Данные**.

Сейчас в связи с дискретными приложениями (геномными, интернетом и т.д.) **происходит перенос в обратную сторону: на графы и БД**.

Самые большие БД – о Живом (биоинформатика, математическая биология). **Их практическая значимость исключительна велика (всё вокруг человека)!**

Чтобы представить, к каким размерам данных нужно готовиться вспомним: один организм определяется последовательностью («геномом») в 4х-буквенном алфавите <длиной 10^9 букв> умножить на <число организмов “10^10”>, которое быстро растёт за счёт секвенирования. Но главное: **ищутся клики в Графе, составленном из всех участков всех геномов**!!

Все геномы вместе – многодольный граф: доля соот-ет геному, вершина в доле – участку в геноме. Ищем клику или плотный подграф:



Как понимать «Технологии обработки БолГрафов»?

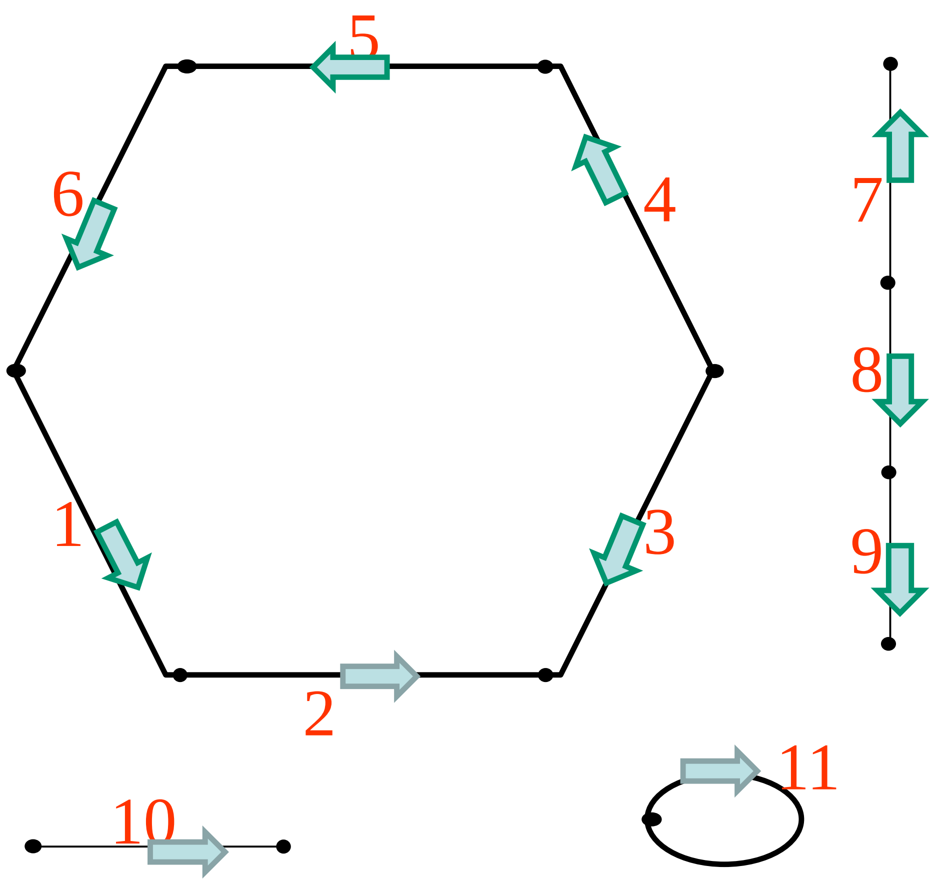
(1) Как хранение и «передача» БГ, средства их «обслуживания», соот-ие программное обеспечение, системное программирование?

(2) **Или как и** методы решения типовых задач о БГ **и** создание списка таких задач?
Содержательная обработка БГ входит в (1)?

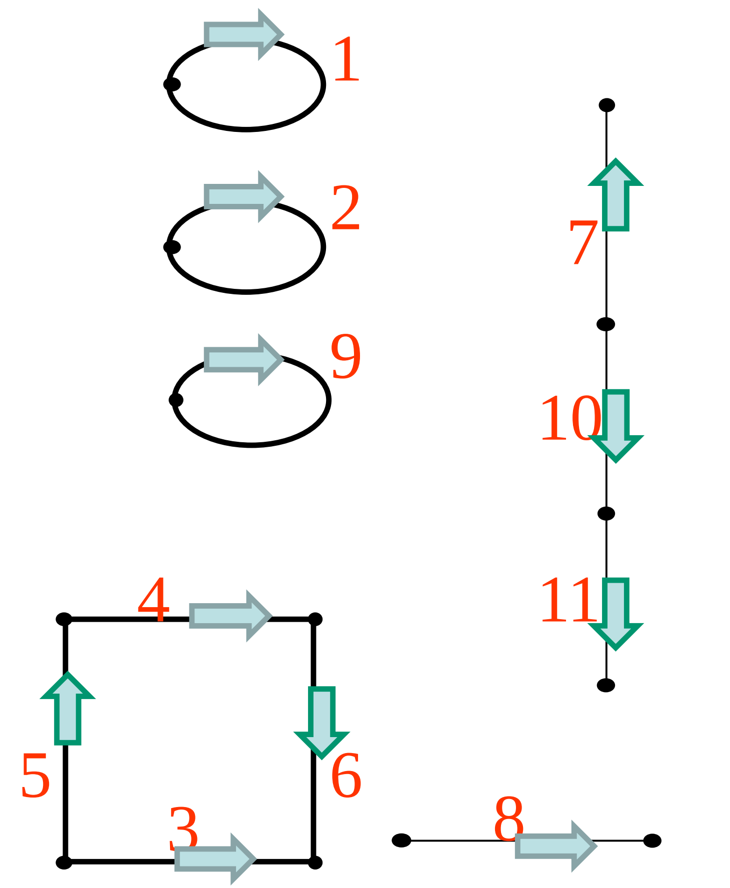
Требования к (1), к организации видов памяти, быстродействиям, допустимым/рекомендуемым метакомандам **определяются из (2)**.

Типичная задача из раздела (2): даны два графа; преобразовать 1й во 2й операциями из фиксированного списка операций:

граф *a*

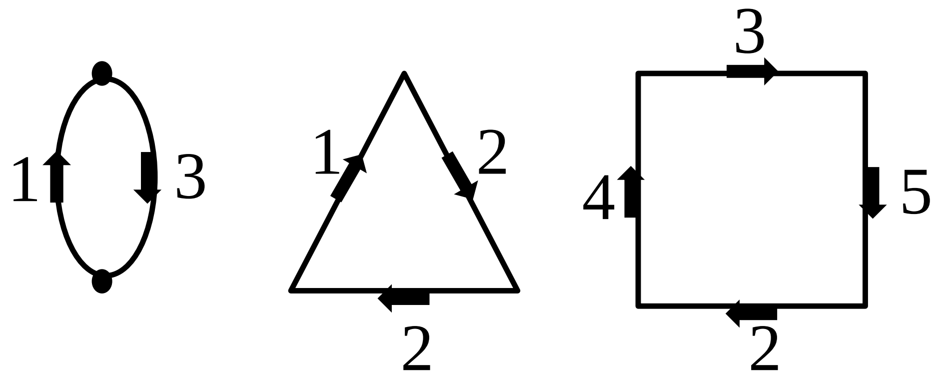


граф *b*

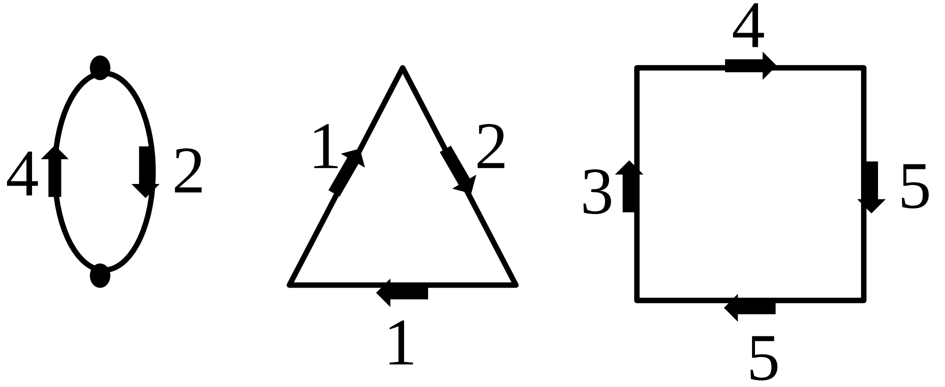


Эта же задача в случае <u>повторения имён рёбер</u>: тогда нужно ещё найти, какие одноимённые в *a* и *b* рёбра соответствуют друг другу так, чтобы минимизировать число операций, преобразующих *a* в *b*:

Граф *a*

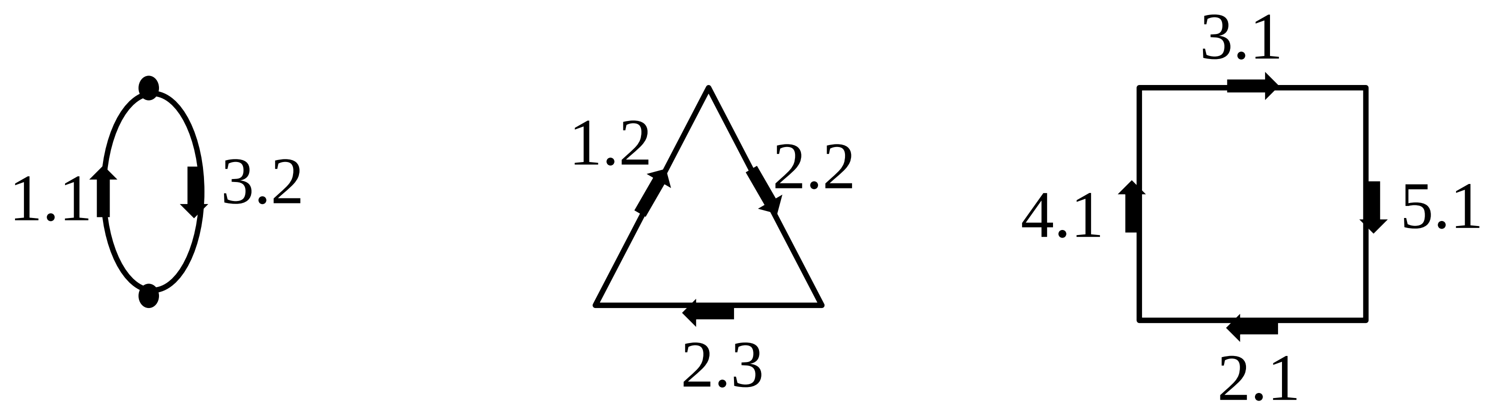


Граф *b*

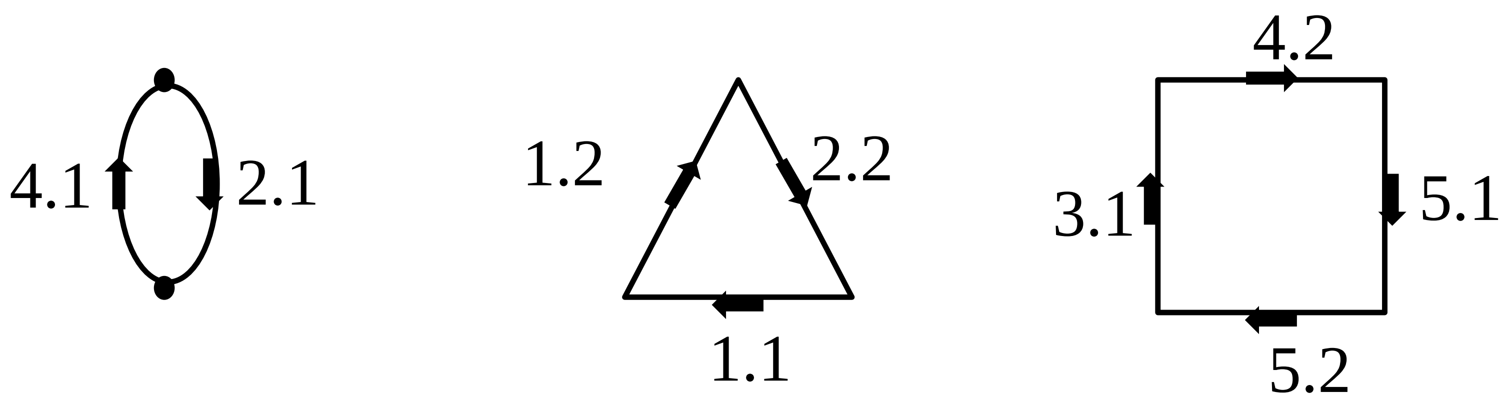


Решение для указанной пары графов:

Граф *a*



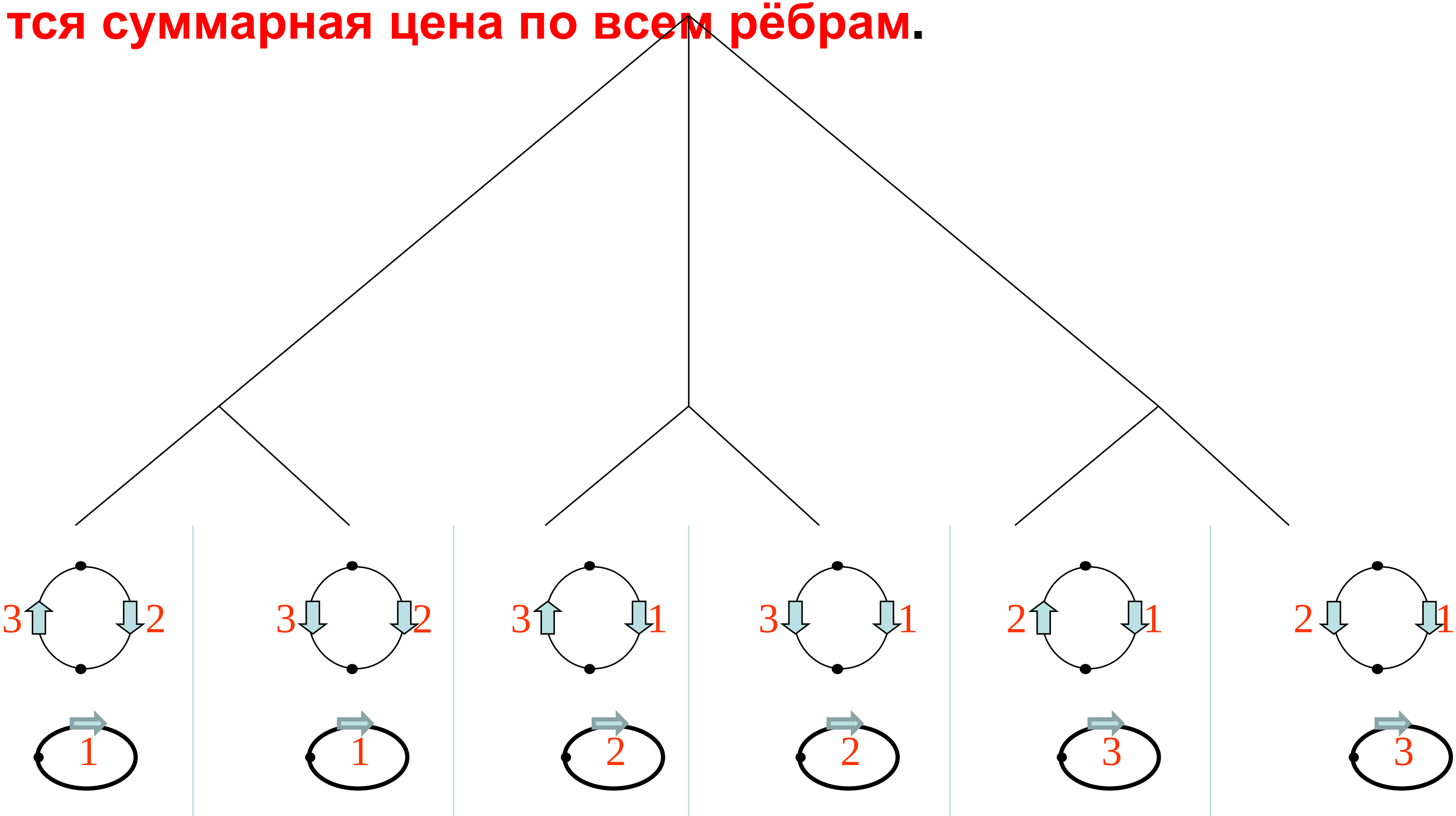
Граф *b*



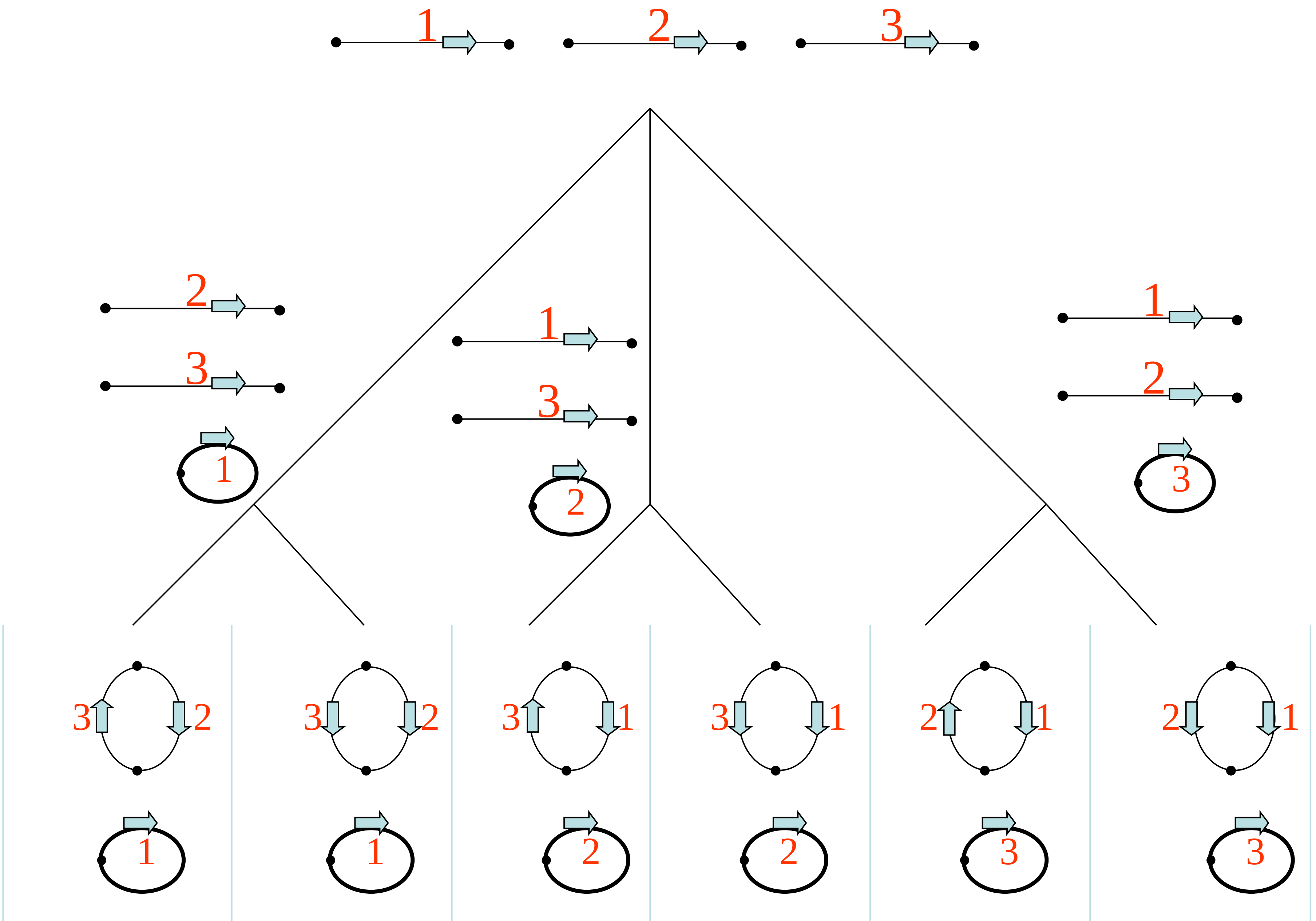
Операции могут естественно иметь ещё и цены: тогда нужно минимизировать суммарную цену искомой последовательности, которая преобразует *a* в *b*.

Продолжить на всё дерево графы, заданные в его листьях, имена не повторяются. На всех рёбрах разрешены операции, преобразующие граф в его начале в граф в его конце

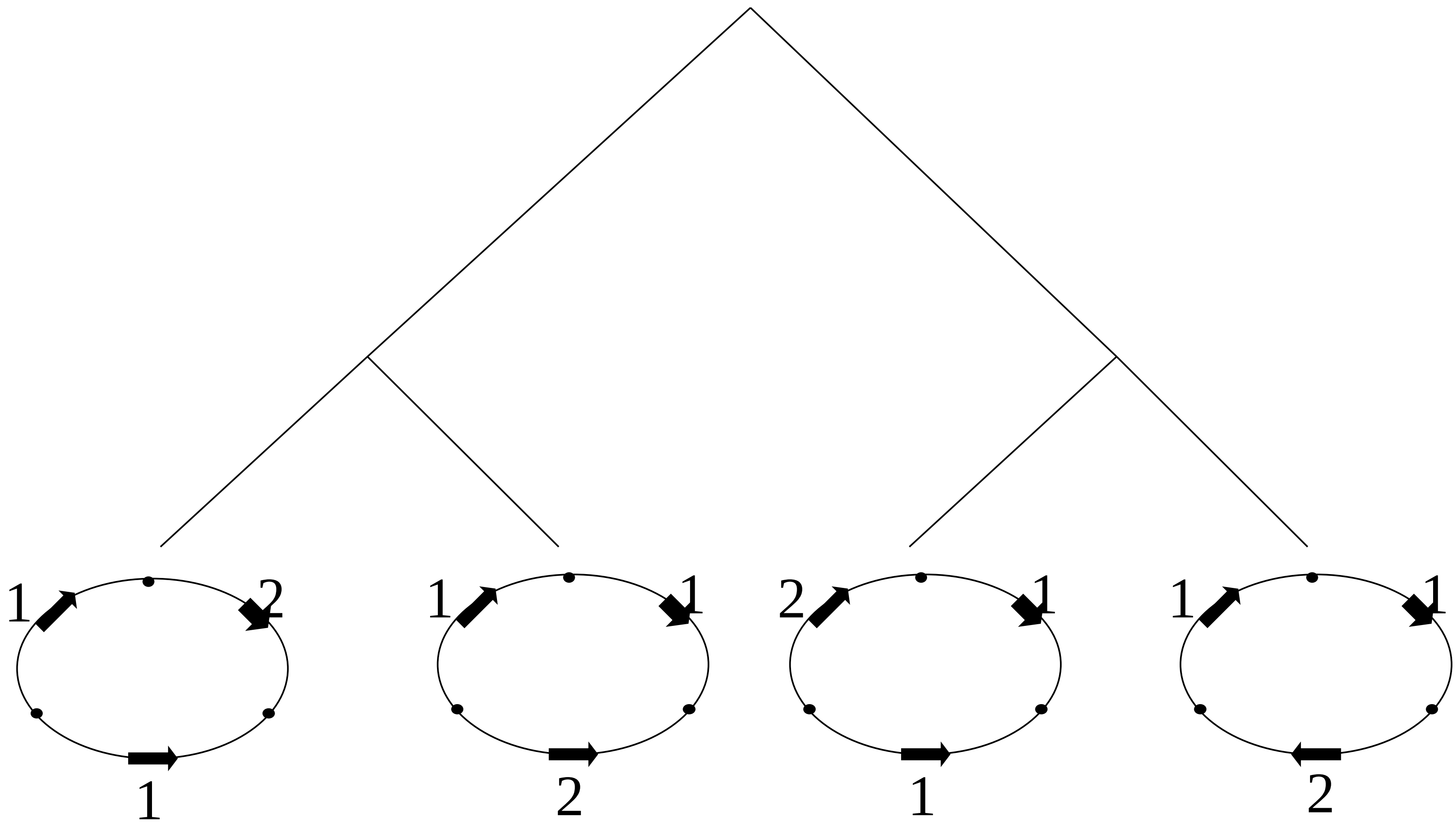
Снова операции могут иметь цены и тогда минимизируется суммарная цена по всем рёбрам.



Решение – графы во внутренних вершинах:



Та же задача, но разрешаются <u>повторения имён во всех графах и вершинах</u>!



NP-трудные!

Нами получены основные результаты

по ЗАДАЧЕ ПРИВЕДЕНИЯ: ***прямой*** алгоритм для случая: повторения имён не допускаются, есть нетривиальные цены. Алгоритм ***сведения*** для случая: повторения имён допускаются, но цены равные: сведение к ЦЛП. Решения одной задачи ДВУМЯ СПОСОБАМИ!

по ЗАДАЧЕ РЕКОНСТРУКЦИИ: ***прямой*** алгоритм для случая: повторения имён не допускаются, есть нетривиальные цены. Алгоритм *сведения* для случая: повторения имён допускаются, но цены равные: сведение к ЦЛП.

Сложность полученных алгоритмов:

ЗАДАЧА ПРИВЕДЕНИЯ: ***прямой*** алгоритм имеет **линейное время и память** от суммарного размера структур. Алгоритм ***сведения***: задача ЦЛП имеет **линейное число переменных и ограничений** в случае циклических структур и **квадратичное число переменных и ограничений** в случае произвольных структур. Сам алгоритм сведения имеет такое же время и память, что и **размер выписываемой** задачи ЦЛП.

ЗАДАЧА РЕКОНСТРУКЦИИ: ***прямой*** алгоритм имеет **кубическое** число переменных и ограничений. Алгоритм ***сведения***: задача ЦЛП имеет кубическое число переменных и ограничений.

Циклический граф состоит из одних циклов.

Паралоги – рёбра с теми же именами.

Рассмотрим далее

циклические графы с **паралогами**

и для них приведём алгоритм сведения к ЦЛП.

Как через ЛП *просто* выразить СлГФ?

Булева переменная z_{kij} равна 1, если паралог *i* гена *k* в структуре *a* соответствует паралогу *j* того же гена *k* в структуре *b*, иначе равна 0; с условием $\sum_i z_{kij} \leq 1, \forall k, j$, аналогично для *j*. Т.е. *z* определяет частичную биекцию паралогов.

Мы доказали: **длина кратчайшей последовательности, приводящй *a* в *b*, равна *B* + S_1 – S_2**, где *B* – число блоков (особых вершин), а в **общем графе *a+b*** S_1 – сумма целых частей половин длин **максимальных по включению участков из обычных рёбер**, S_2 – число циклов, состоящих из обычных рёбер.

Нужно в задаче ЦЛП вычислить ***B* + S_1 – S_2**. Начнём с произвольной нумерации паралогов.

Задача приведения *a* к *b* сводится к задаче приведения общего графа *a+b* к финальному виду.

<u>Определение</u> общего графа *a+b*:

Обычная *вершина* в ***a+b*** – край i_j представленный в *a* и *b*.

Особая *вершина в* ***a+b*** – максимальный по включению связный участок из особых генов.

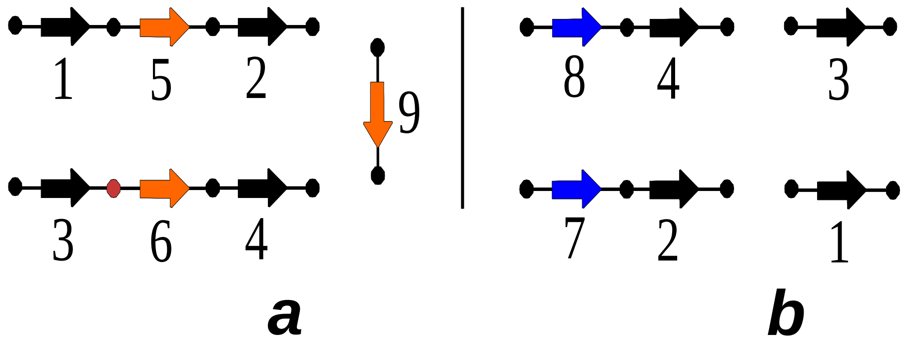
Обычное ребро ***a+b*** *соединяет* обычные вершины, если они склеены в ***a* или *b***, и помечается именем структуры.

Особое ребро соединяет особую вершину с обычной, если край блока склеин с краем общего гена в ***a* или *b*.**

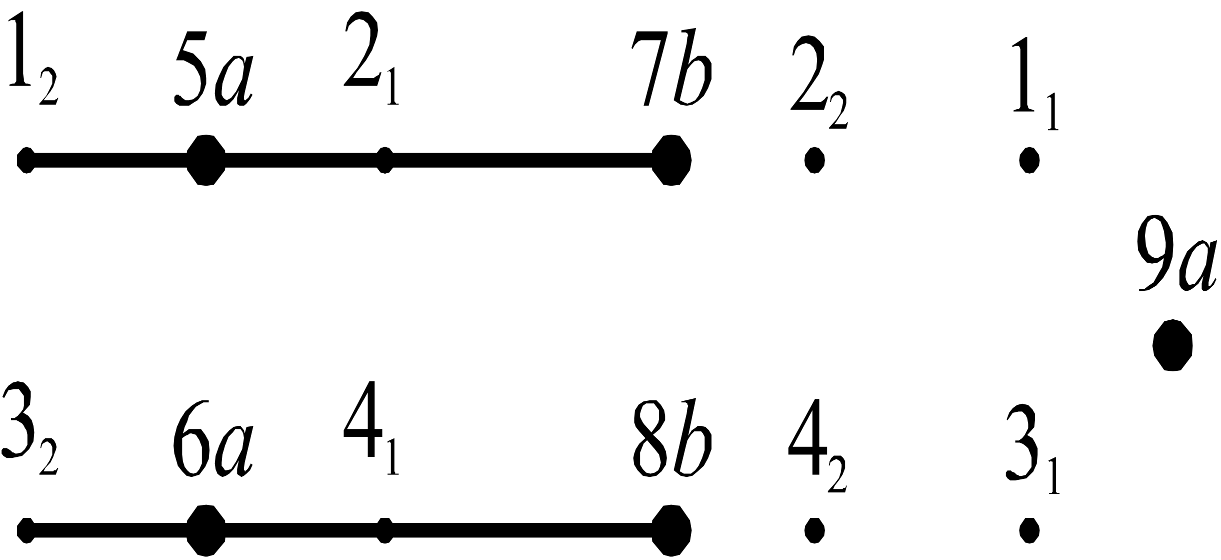
Общий граф ***финального вида***, если он общий граф двух совпадающих структур, т.е. вида *c*+*c*.

Пример общего графа *a+b* :

Исходные *a* и *b*:



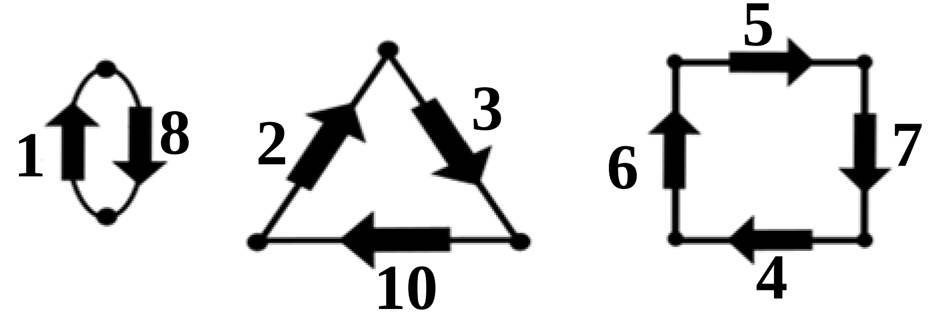
Их общий граф *a+b*:



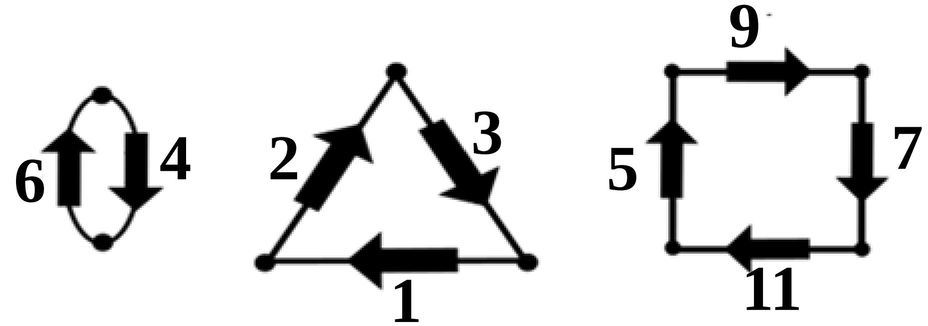
В *a* и *b* особые гены помечены цветом, в *a+b* им соответствуют большие кружочки. **Общий граф** состоит из циклических и линейных *компонент* и изолированных вершин.

Пример общего графа *a*+*b* для циклического случая:

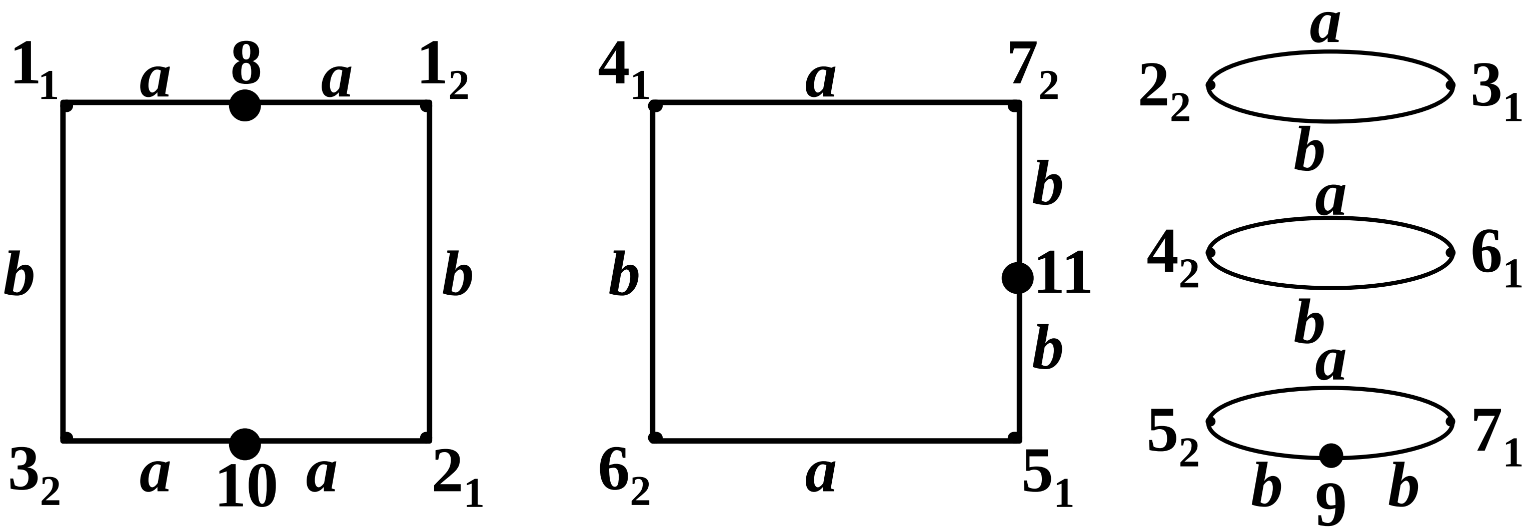
a:



b:



a+*b*:



Как через ЛП выразить сложный граф?

Булева переменная z_{kij} равна 1, если паралог *i* гена *k* в структуре *a* соответствует паралогу *j* того же гена *k* в структуре *b*, иначе равна 0; с условием $\sum_i z_{kij} \leq 1, \forall k, j$, аналогично для *j*. Т.е. *z* определяет частичную биекцию паралогов.

Мы доказали: **длина кратчайшей последовательности, приводящй *a* в *b*, равна $B + S_1 - S_2$**, где **B – число блоков** (особых вершин), а в **общем графе *a+b*** S_1 – сумма целых частей половин длин **максимальных по включению участков из обычных рёбер**, S_2 – число циклов, состоящих из обычных рёбер.

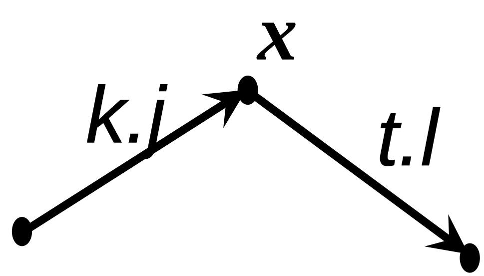
Нужно в сформулировать задачу ЦЛП, в которой вычисляется
$B + S_1 - S_2$.
Начнём с произвольной нумерации паралогов.

Число *B* блоков = особых вершин:

Число *B* блоков = половине **числа склеек в *a* и *b* особых и общих рёбер**. **Каждой вершине сопоставим булеву переменную *x* с условием**

$$x \geq \sum_{j} z_{kij} - \sum_{s} z_{tls}$$

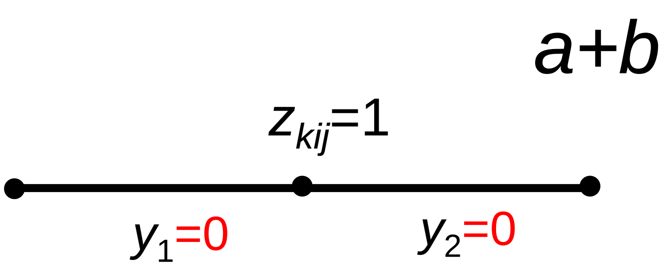
и симметричное неравенство (*j* и *s* меняются местами).

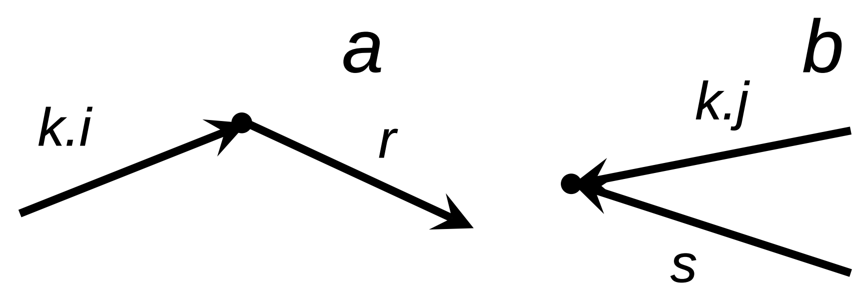


Тогда $B = 0.5 \cdot \sum x$. И ещё учёт циклических блоков.

Число S_1 – сумма целых частей половин длин максимальных по включению участков из **обычных** рёбер в a+b:

Рассмотрим два соседних ребра из такого участка. Они соединяются в однотипном крае z-соответствующих рёбер $k.i$ и $k.j$. Последние соединяются с какими-то r и s в a и b, соответственно. Эти r и s – z-общие, так как исходные дальние вершины обычные. Общность r и s означает, что обе скобки равны 1, как и первое слагаемое. Это означает оба y_1 и y_2 не ≠ 0 одновременно.

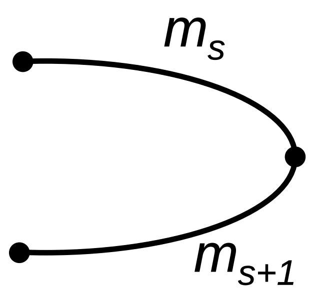




Условие $y_1+y_2 \geq z_{122}+(z_{rr1}+z_{rr2})+(z_{ss1}+z_{ss2})-2$. Тогда $\boldsymbol{S}_1$ $= \sum y$ в точке min

Число S_2 – числа циклов, состоящих из **обычных** рёбер в $a+b$: каждому такому ребру припишем целочисленную переменную $u_s \le m_s$, где m_s соответственно равно 1,2,…, а булева переменная p_s имеет ограничение $p_s \le u_s/m_s$.

Условия $u_s \le u_{s1} + m_1(1–z_{122})$, $u_{s1} \le u_s + (1–z_{122})$ обеспечивает равенство всех соседних u_s, т.е. всех u_s вдоль каждого цикла. На **любом цикле** не более одного p_s равно 1, а в точке минимума ровно одно $p_s = 1$.



$S_2 = \Sigma p$. Минимизируется функция $F = 0.5\cdot\Sigma x + \Sigma y – \Sigma p$.