Институт проблем передачи информации

Российской академии наук

Лаборатория «**Математических методов и моделей в биоинформатике**»

сайт лаборатории: **http://lab6.iitp.ru/**

адрес: **lyubetsk@iitp.ru**

В.А. Любецкий

МАТЕМАТИЧЕСКИЕ и COMPUTER SCIENCE ПРОБЛЕМЫ БИОИНФОРМАТИКИ

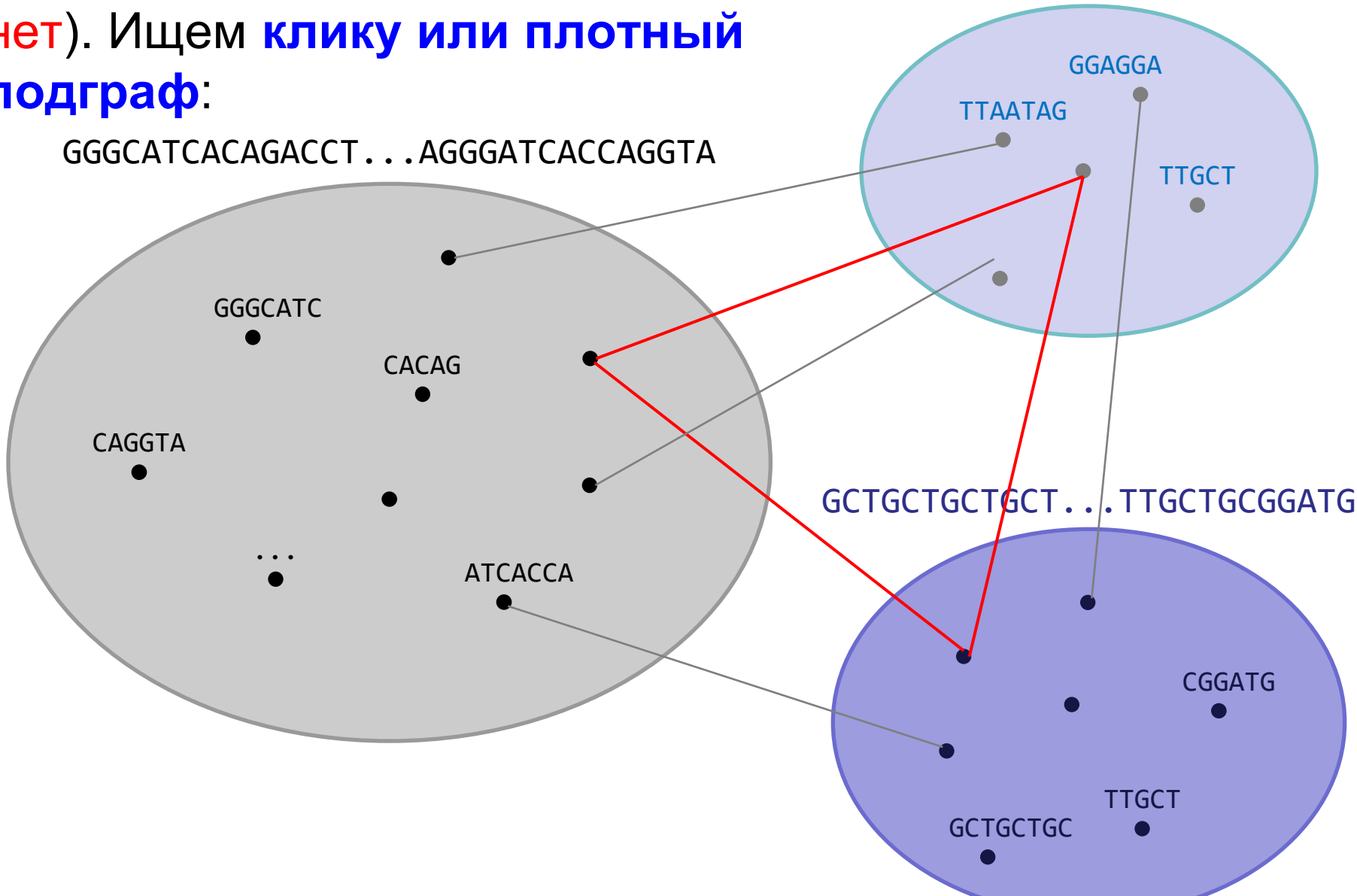
Тезис: «Любое ЖИВОЕ следует изучать с помощью математических моделей и алгоритмов (программ, суперкомпьютеров, баз данных, больших данных и т.п.)», т.е. с помощью математики и CS. ????

В части Живого самые большие: сфера применений, ресурсы, финансирования, объёмы данных, количества институтов, журналов и т.д.

О роли графов. (динамич. системы – блуждание по графу)

Типичная методика (как и в физике, например): биологическая проблема → математическая и CS проблема, программирование, счёт → и назад.

0) Самая типичная задача: дан **многодольный граф**, ребро – близость или иная характеристика пары слов на его концах (внутри доли рёбер нет). Ищем **клику или плотный подграф**:



00) Вторая по типичности задача: кластеризация данного множества (последовательностей в 4х или 20ти буквенном алфавитах).

Это – классическая задача; как всегда, трудно сформулировать свойство, по которому последовательности объединяются в кластер (тут специфика биоинформатики). Для решения задачи кластеризации известны хорошие алгоритмы.
Но она далека от окончательного решения.

Кластеризации в многодольном графе состоит в разбиении графа на подграфы: в каждом подграфе (кластере) как можно **меньше вершин из одной доли** и как можно **больше долей представлено (т.е. размер подграфа как можно больший)**.

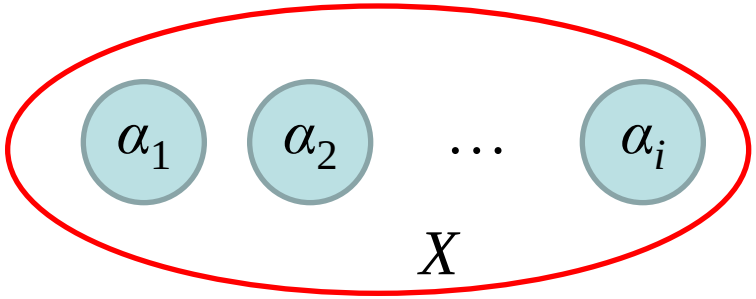
Задача 2х-параметрическая – обычная трудность!

Нашей группой (Л. Рубанов, А. Селиверстов, К. Горбунов, О. Зверков и др.) для этой и всех последующих задач/проблем предложены алгоритмы, обычно с доказательствами точности, линейные или близкие к ним.

Но нужны лучшие!

Математические и computer science проблемы биоинформатики:

1) Даны *n* натуральных чисел с повторениями, можно ли разбить их множество на две части с одинаковыми суммами чисел из каждой части:



$$\exists X \; ?\left(\sum_{i \in X} \alpha_i = \sum_{j \in CX} \alpha_j\right)$$

Полный перебор требует 2^n шагов, нужен полином от *n*

Задача близка к задаче поиска ближайшей к данной поверхности вершины единичного куба в *n*-ом пространстве.

2) Множественное выравнивание данных последовательностей: глобальное и локальное.

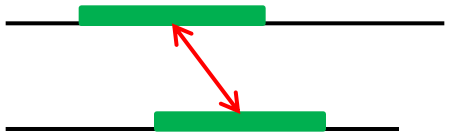
Парное выравнивание (две последовательности)

```
 CGTAATAGGACTATGA      =>    CGTAATA-GGACT--ATGA
AGTATACGGACTTAATGC     =>    AGTA-TACGGACTTAATGC
                              ***  **  *****   ***
```

ГЛОБАЛЬНОЕ: от начала до конца обоих последовательностей:
Функционал: расстояние Левенштейна (min) или score (max), с постоянным или аффинным штрафом за делецию. Качество выравн. зависит от функц-ла.
Матрицы близости букв: PAM, BLOSUM. Точные алгоритмы (ДП): Needleman-Wunsch 1970: $O(mn)$, $O(mn)$. Hirschberg 1975: memory $O(\min\{m,n\})$, time $O(mn)$.

ЛОКАЛЬНОЕ: проблема в выборе оптимальных начал и концов БОКСОВ для их глобального выравнивания:

Точные алгоритмы (ДП): Smith-Waterman 1981: $O(mn)$, $O(m^2n)$; Gotoh 1982, Altschul 1986: $O(mn)$, $O(mn)$; Myers-Miller 1988: $O(\min\{m,n\})$, $O(mn)$. Не реалистичны для больших m, n и более 1 бокса.



Эвристические алгоритмы на основе выделения похожих коротких слов:
BLAST – одна последовательность произвольная короткая (query), другая – длинная, из предварительно обработанного набора (database).
LASTZ – локальное выравнивание двух близкородственных полных геномов (обе последовательности длинные).

Множественное выравнивание (*k* посл-тей длины ≤*n*)

ГЛОБАЛЬНОЕ: каждая последовательность от начала до конца.
ЛОКАЛЬНОЕ: выбор оптимальных начал и концов БОКСОВ для их глобального выравнивания в *ПОЧТИ* каждой последовательности.

Функционал не очевиден, например: сумма/max/средний попарный score, коэффициент идентичности всех позиций выравнивания…

- Как учитывать штраф за делецию буквы в одной или более послед-тях?
- Как назначать штраф за выбрасывание последовательности?

Результат очень сильно зависит от функционала.

Точные алгоритмы – *k*-мерное расширение парного выравнивания (используется *k*-мерная scoring matrix) алгоритмами ДП типа NW и SW. Сложность: $O(n^k)$, $O(n^k)$ - в общем случае, с оптимизацией по Хиршбергу потенциально достижимо $O(n^{k-1})$.
Практически применимы только при $k < 10$. Пакет MSA (NCBI).

Эвристические алгоритмы реализуются популярными программами, включая общедоступные облачные сервисы, основанные на этих программах:
Clustal, T-COFFEE, MUSCLE, MAFFT, MEME…

Итак: весьма трудоёмкие алгоритмы, функционал не всегда задан, не гарантируется достижение экстремума, ориентированы в основном на близкие последовательности (гомологи), сводят проблему к глобальному множественному выравниванию. **ВАЖНАЯ ПРОБЛЕМА!**

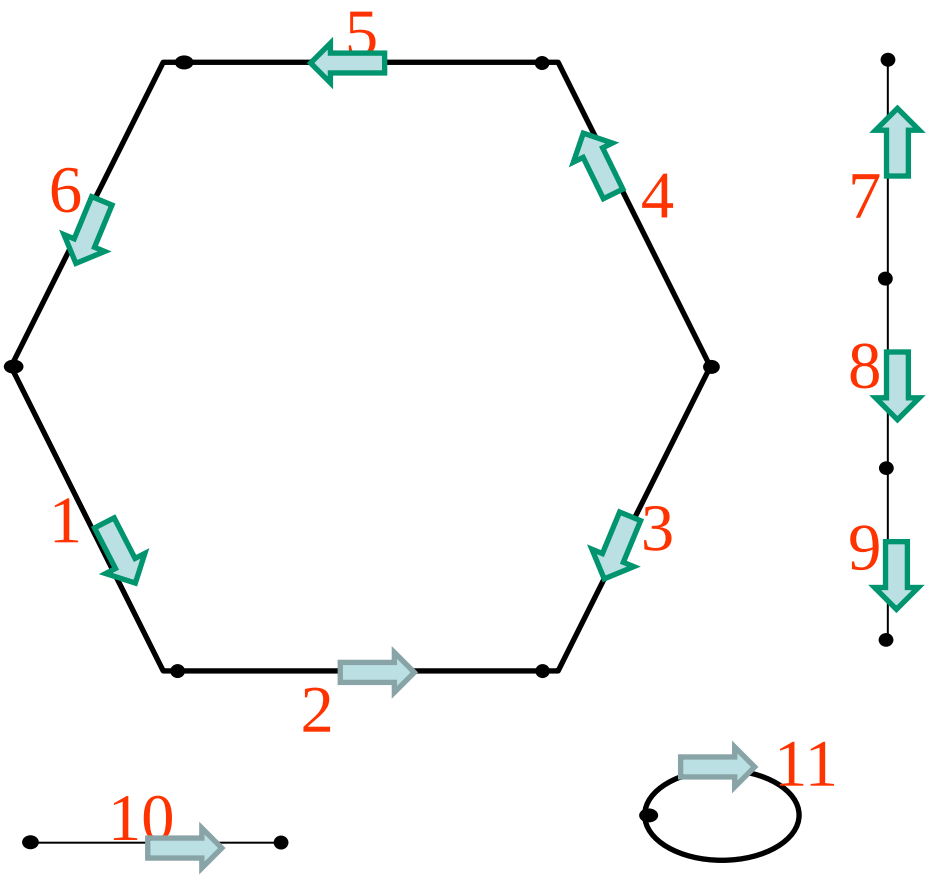
3) Даны графы *a* и *b* и дан фиксированный список естественных операций над графами. Какова кратчайшая последовательность этих операций, преобразующая *a* в *b* (= **задача преобразования одного графа в другой**).

Каждой **операции назначена своя цена**, **найти цену самой дешёвой последовательности операций** которые переводят граф *a* в граф *b*, и **саму эту последовательность**.

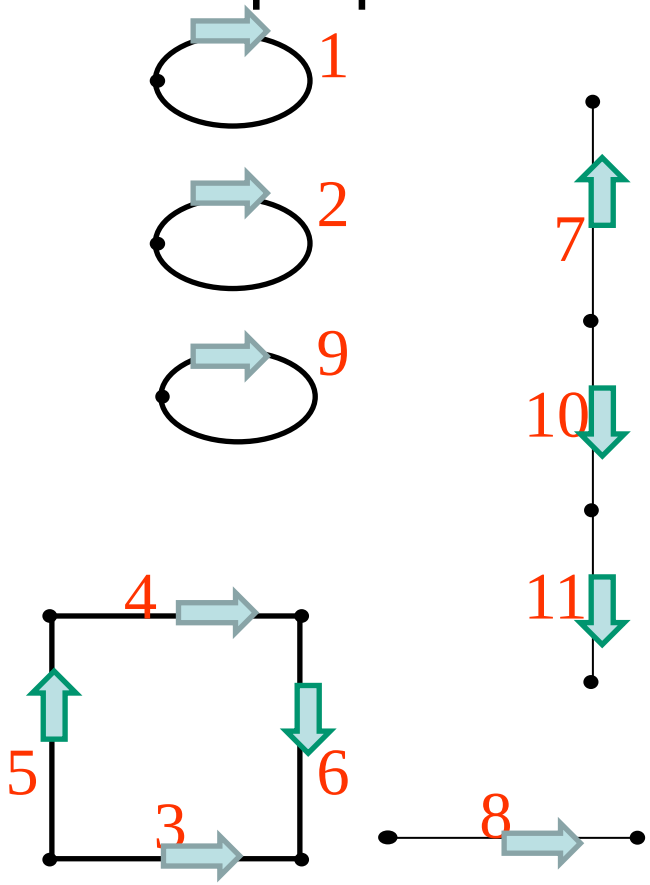
Эта цена и есть **РАССТОЯНИЕ** между данными графами. Определить такое расстояние ещё одна проблема.

Показаны упрощённые графы типичные для биоинформатики, они состоят только из циклов и цепей:

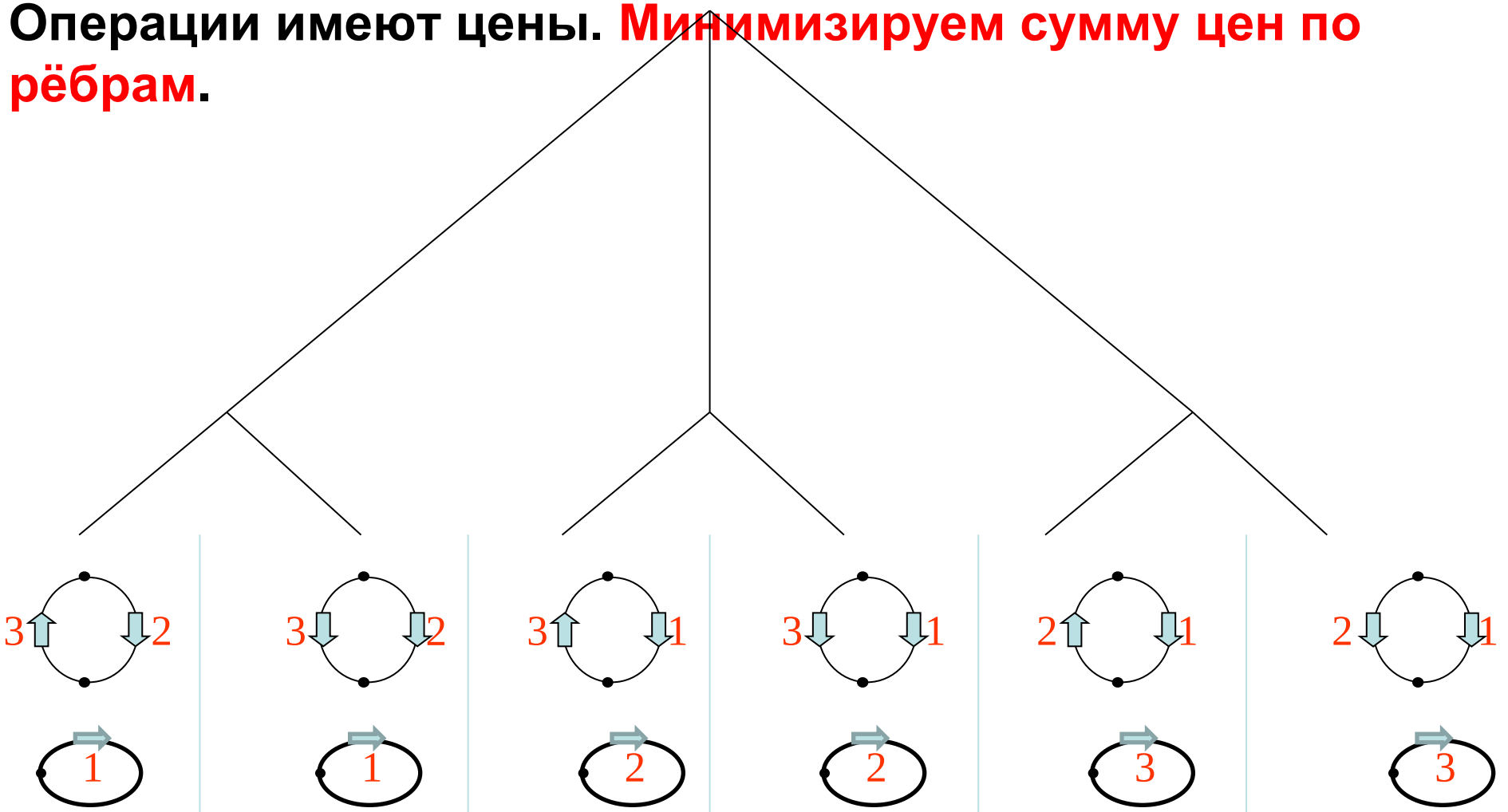
граф *a*



граф *b*



4) Оптимальное продолжение на всё дерево графов, заданных в его листьях = проблема реконструкция.
Расстановка – каждой внутренней вершине приписан свой граф. На всех рёбрах разрешены операции, преобразующие граф в его начале в граф в его конце
Операции имеют цены. Минимизируем сумму цен по рёбрам.



5) Большие задачи целочисленного линейного программирования (ЦЛП), решающие эти проблемы. Как их считать??

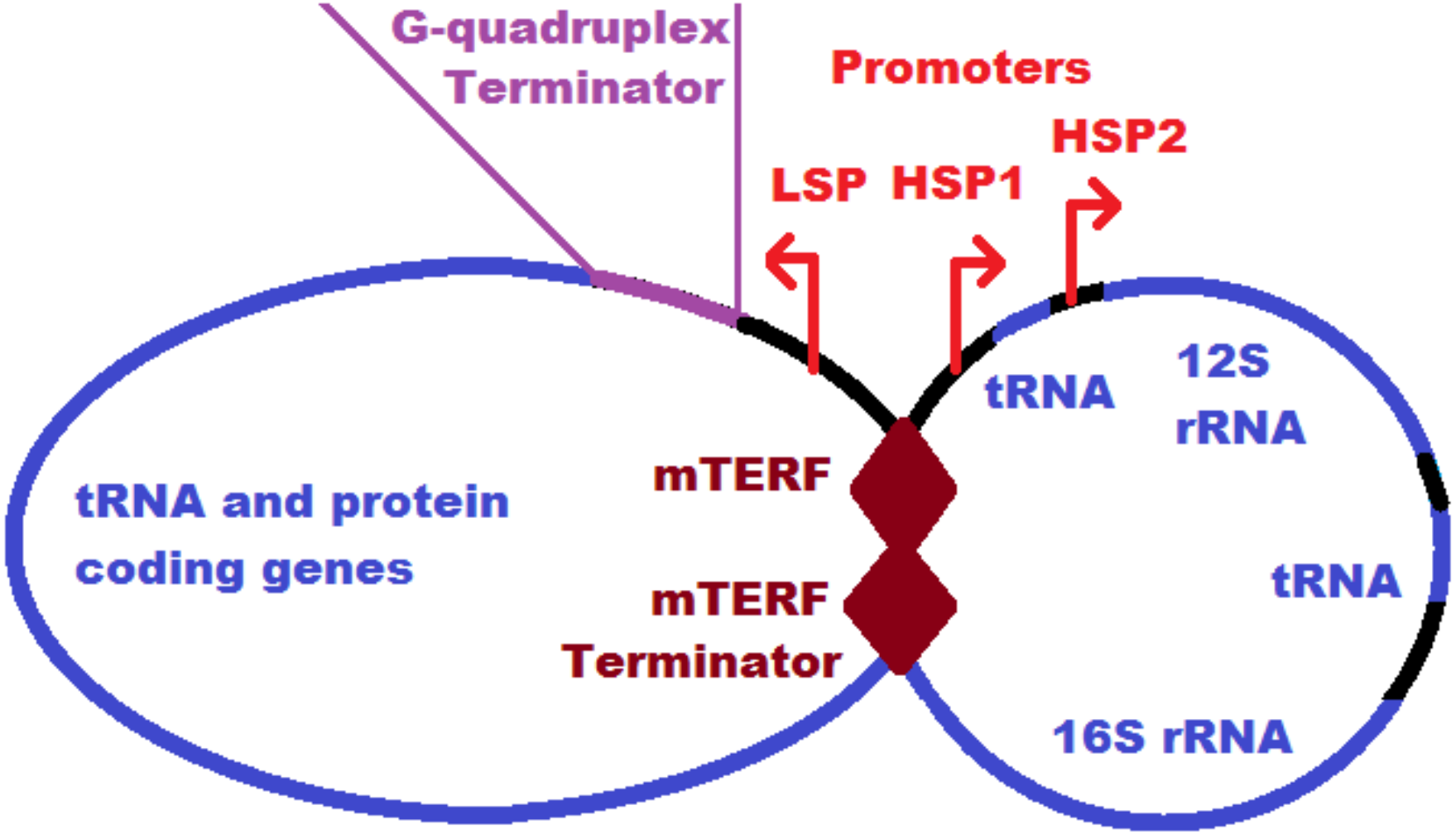
6) Что такое средний граф для данного набора графов. Иными словами: **задача согласования большого числа деревьев в единое дерево.**

Нужны определение и эффективный алгоритм!

7) Описать смену режимов в функционировании динамических систем определённых типов (важных в биологии). Мы приведём три примера таких систем.

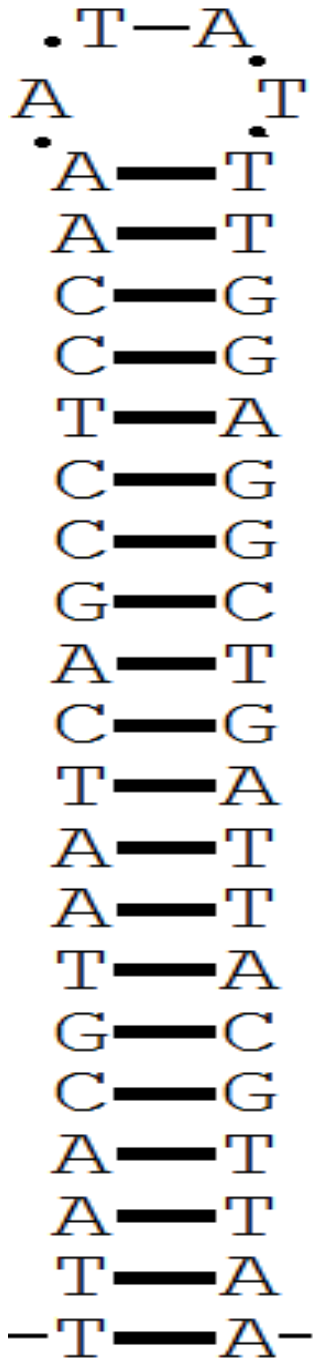
***7a)*. Конкуренция встречных потоков**.

Встречные потоки вдоль кривой с заданными участками въезда и препятствиями; вычисление интенсивности движения на заданном участке кривой. В среде вокруг кривой плавают тысячи одинаковых «машин» (=молекул). Они толкаются, чтобы связаться со «входами на дорогу» LSP и HSP.



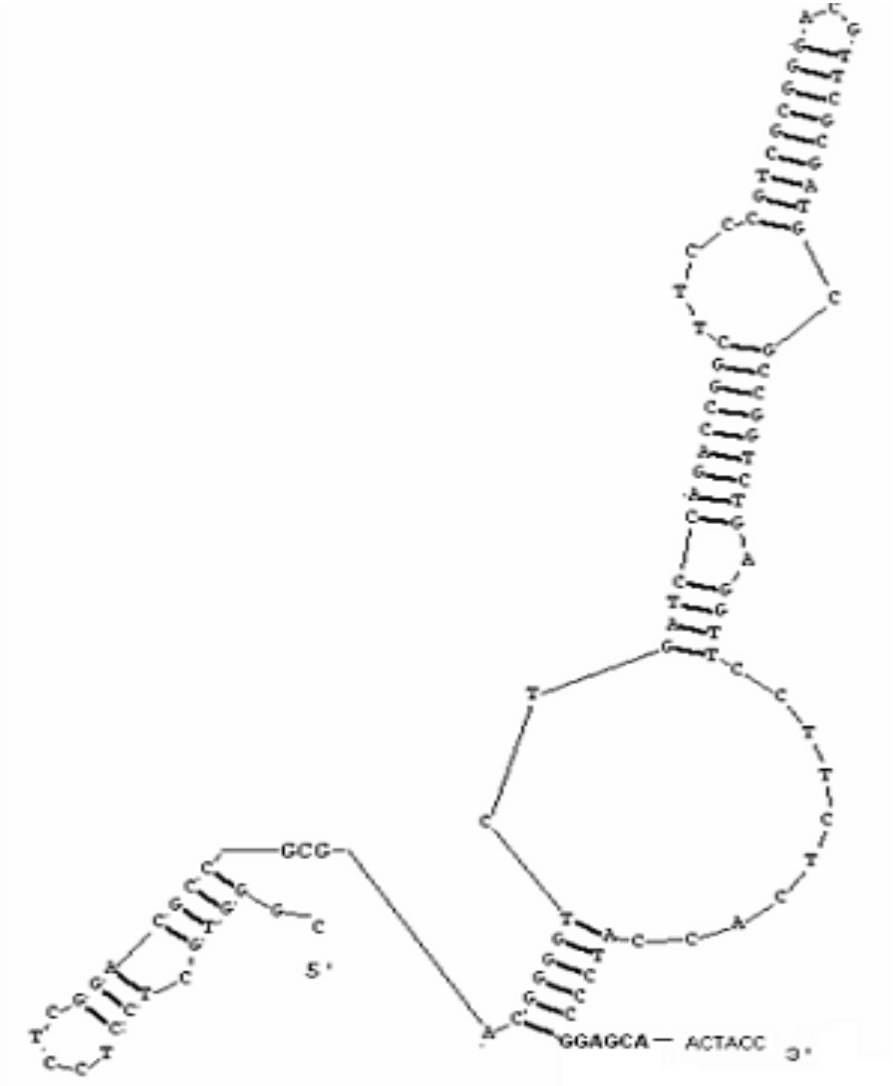
7***b***). **Преследование одной машины другой: 1я машина в «вязкой среде», которую создаёт 2я машина, но на неё среда не влияет**.

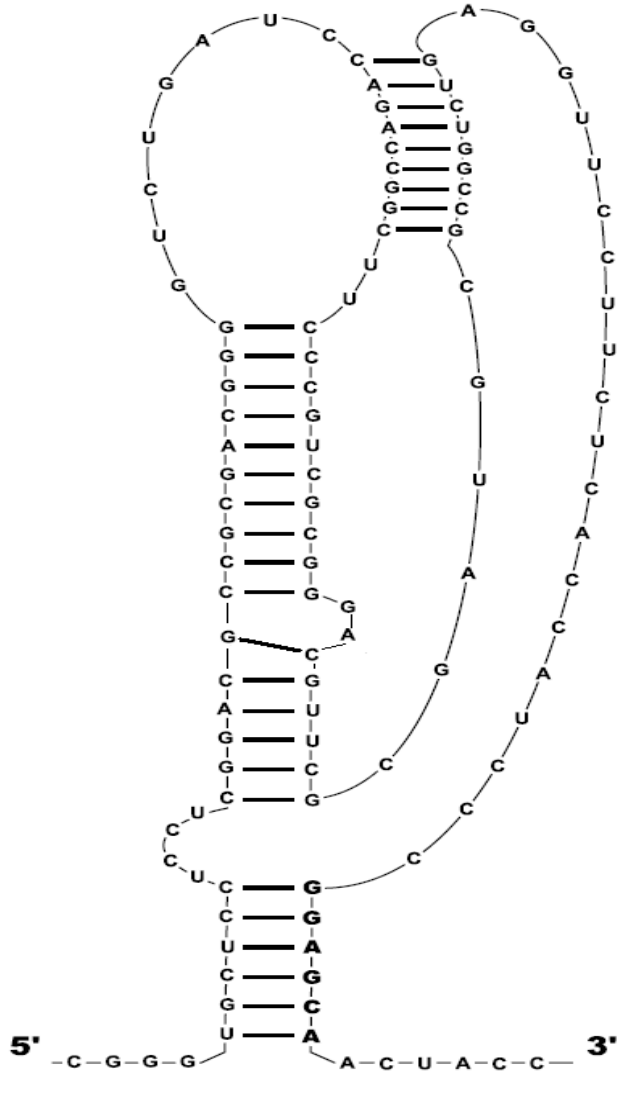
«Дорога», по которой друг за другом едут две машины, – последовательность в алфавите {A,C,T,G}. На участке между машинами возникает **среда**: **участок изгибается и буквы A и T, C и G склеиваются между собой.** Такая склейка называется шпилькой.



TTAACGTAATCAGCCTCCAAATATTTGGAGGCTGATTACGTTAA

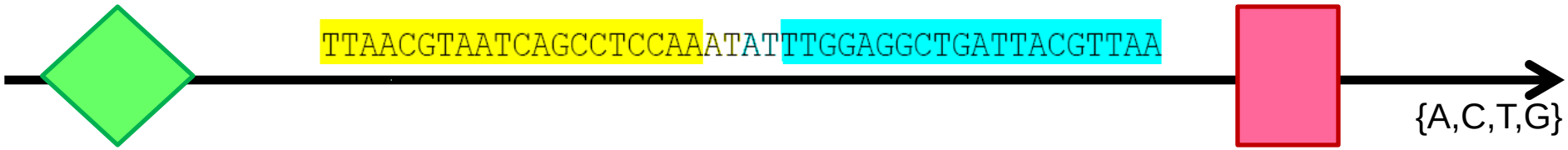
На участке образуется много шпилек! Они быстро сменяют друг друга. Получается среда – быстро меняющиеся множество шпилек (среда дышит как пена). Шпильки пытаются сдёрнуть с дороги 1ю машину, но не влияют на 2ю.



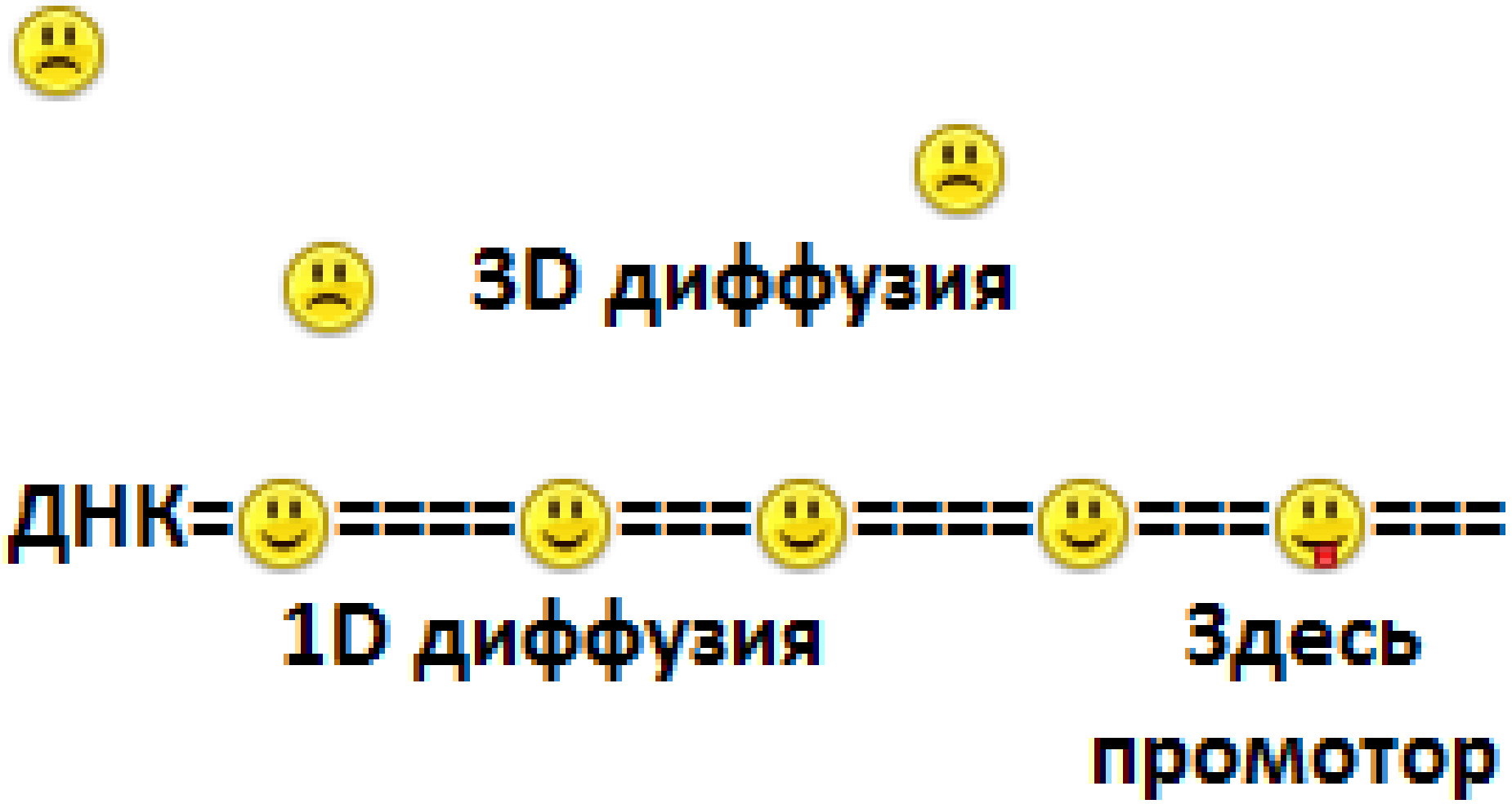


Сам участок перемещается, так как он зажат между двумя машинами, ползущими по последовательности друг за другом. Движение машин асинхронно, по своим заданным законам, а между ними дышит среда из шпилек (=вторичная структура).

Все это нами моделируется, сложное компьютерное моделирование.



7*c*) Сочетание трёхмерной и одномерной диффузий:



Во всех случаях нужно найти низкой степени полиномиальный (практически – линейный или близкий к нему) вычислительный алгоритм, который решает проблему.

Различие между экспоненциальным и полиномиальным (скажем, 1000й степени) не так важно.

Важную роль в исследовании играют программирование для суперкомпьютеров (особенно для систем с распределённой памятью) и создание/использование больших данных; а также – организация памяти вычислительного устройства и распределение промежуточных задач между вычислительными процессами.

Важно понижение размерности в соответствующей задаче ЦЛП.

Обычно не удаётся решить проблему точным алгоритмом, поэтому большое значение приобретают моделирование, как и сравнение многих эвристических алгоритмов решения одной и той же проблемы.

Методы поиска информации в больших базах данных или DATA MINING биологической (или иной!) информации.

Примеры используемых в нашей предметной области баз/банков данных –

GenBank (менее формализованная),

Ensembl (более формализованная).

Некоторые наши публикации 2016 года,
хотя все публикации можно найти на странице http://lab6.iitp.ru/ :

V.A. Lyubetsky,
R.A. Gershgorin, A.V. Seliverstov, K.Yu. Gorbunov,
Algorithms for Reconstruction of Chromosomal Structures,
BMC Bioinformatics, 2016, vol. 17, no. 40.

L.I. Rubanov, A.V. Seliverstov, O.A. Zverkov, V.A. Lyubetsky, A Method for Identification of Highly Conserved Elements and Evolutionary Analysis of Superphylum Alveolata,
BMC Bioinformatics, 2016, vol. 17, no. 385.

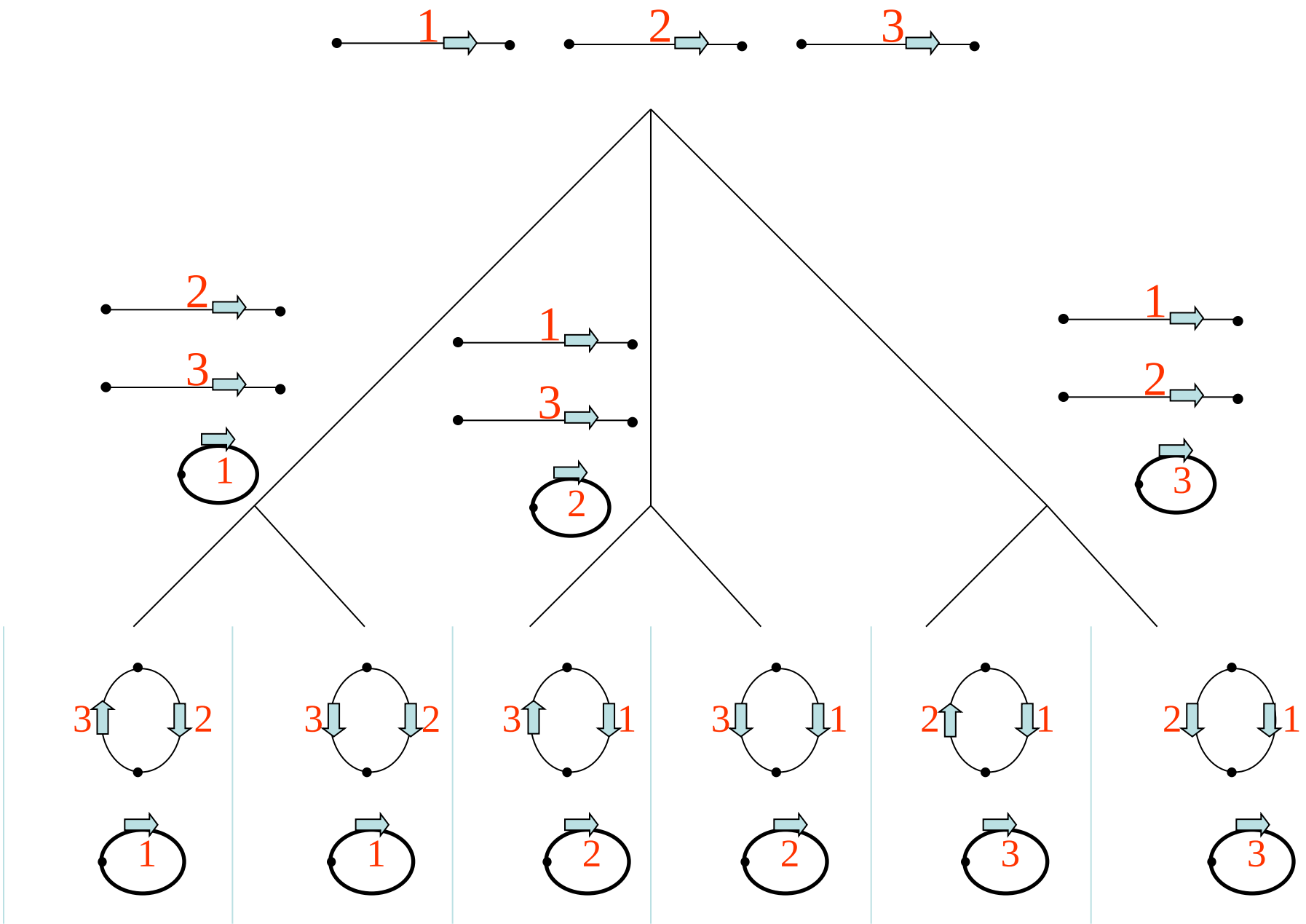
O.A. Zverkov, A.V. Seliverstov, V.A. Lyubetsky, Regulation of Expression and Evolution of Genes in Plastids of Rhodophytic Branch,
Life, 2016, vol.6, no. 7.

СПАСИБО за внимание

Нами найден **линейный и точный! алгоритм** при одном наборе операций.

Исследование этой задачи для других наборов операций представляет большой интерес.

Пример: решение – графы во внутренних вершинах:

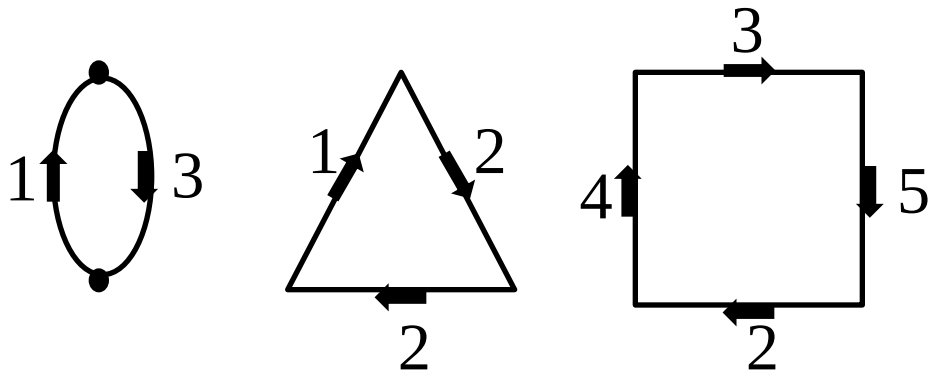


Нами найден кубический точный! алгоритм, который минимизирует цену расстановки в этой задаче реконструкции.

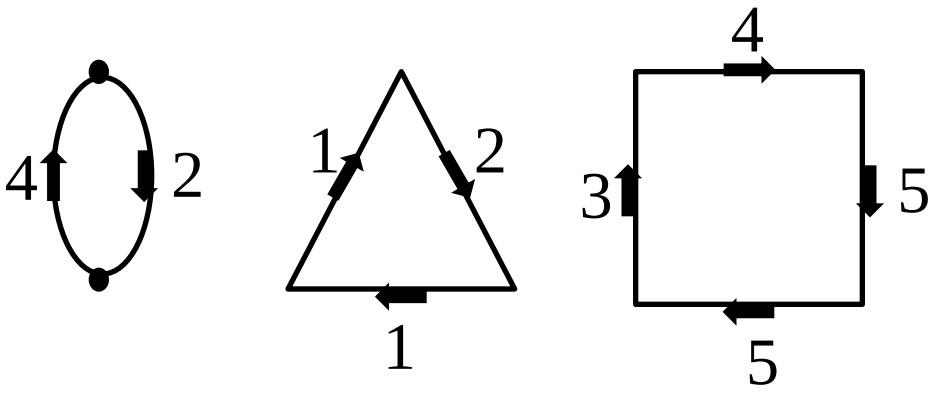
Сведение задачи вычисления преобразования и расстояния между графами в случае повторения имён рёбер к задаче ЦЛП.

Даны:

Граф *a*



Граф *b*

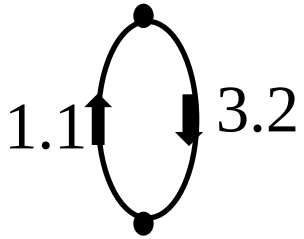


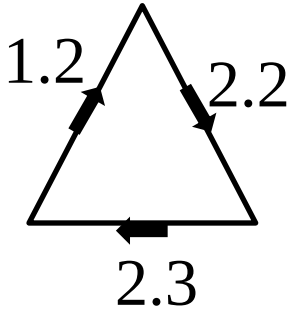
$$F = 0.5\sum x + \sum y - \sum p$$

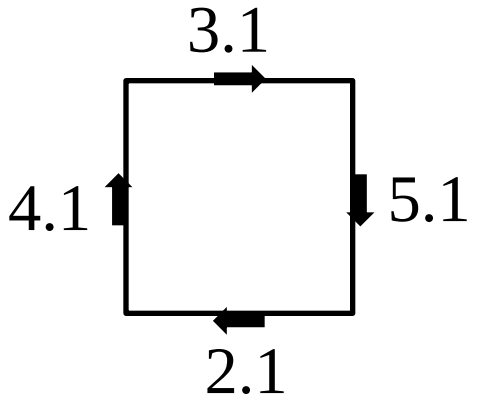
Нужно: нумеровать одноимённые в *a* и *b* рёбра, чтобы минимизировать число операций, преобразующих *a* в *b*. Решение – *минимизировать ц.ф. F , которую мы указали*!

Решение для исходной пары графов:

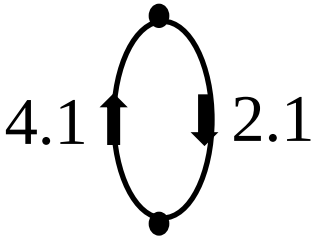
Граф *a*

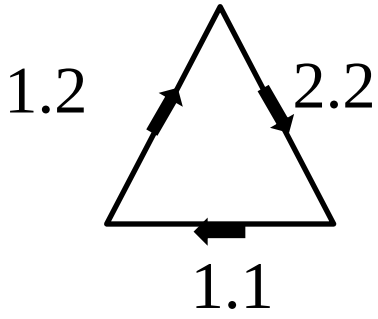


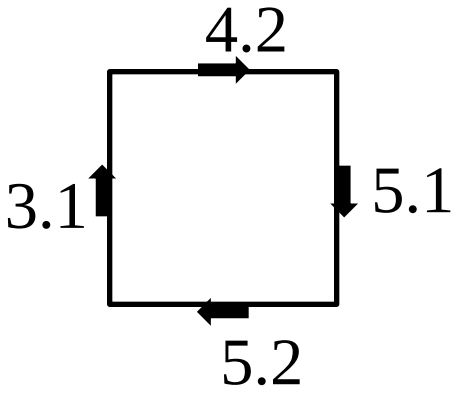




Граф *b*

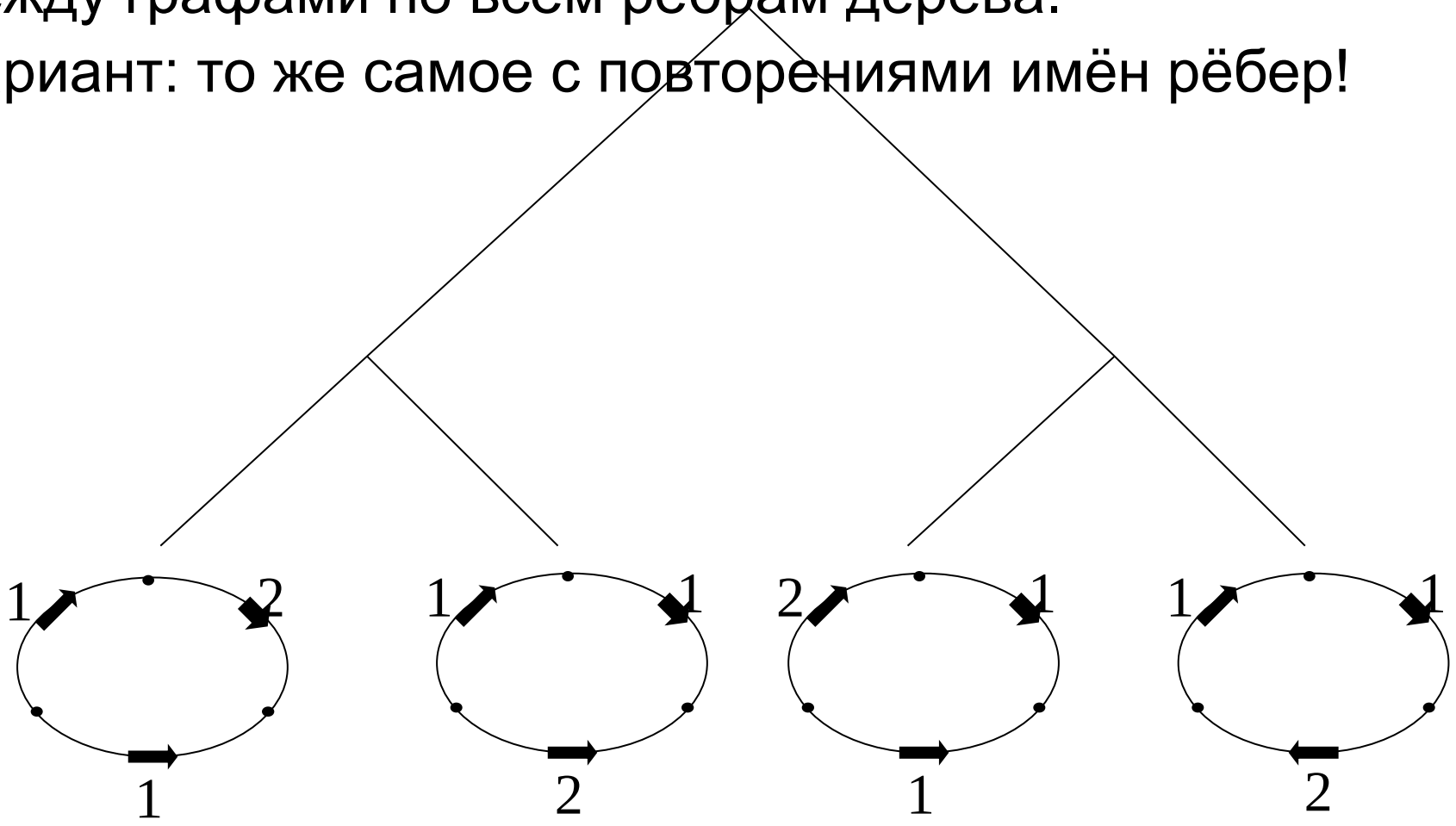




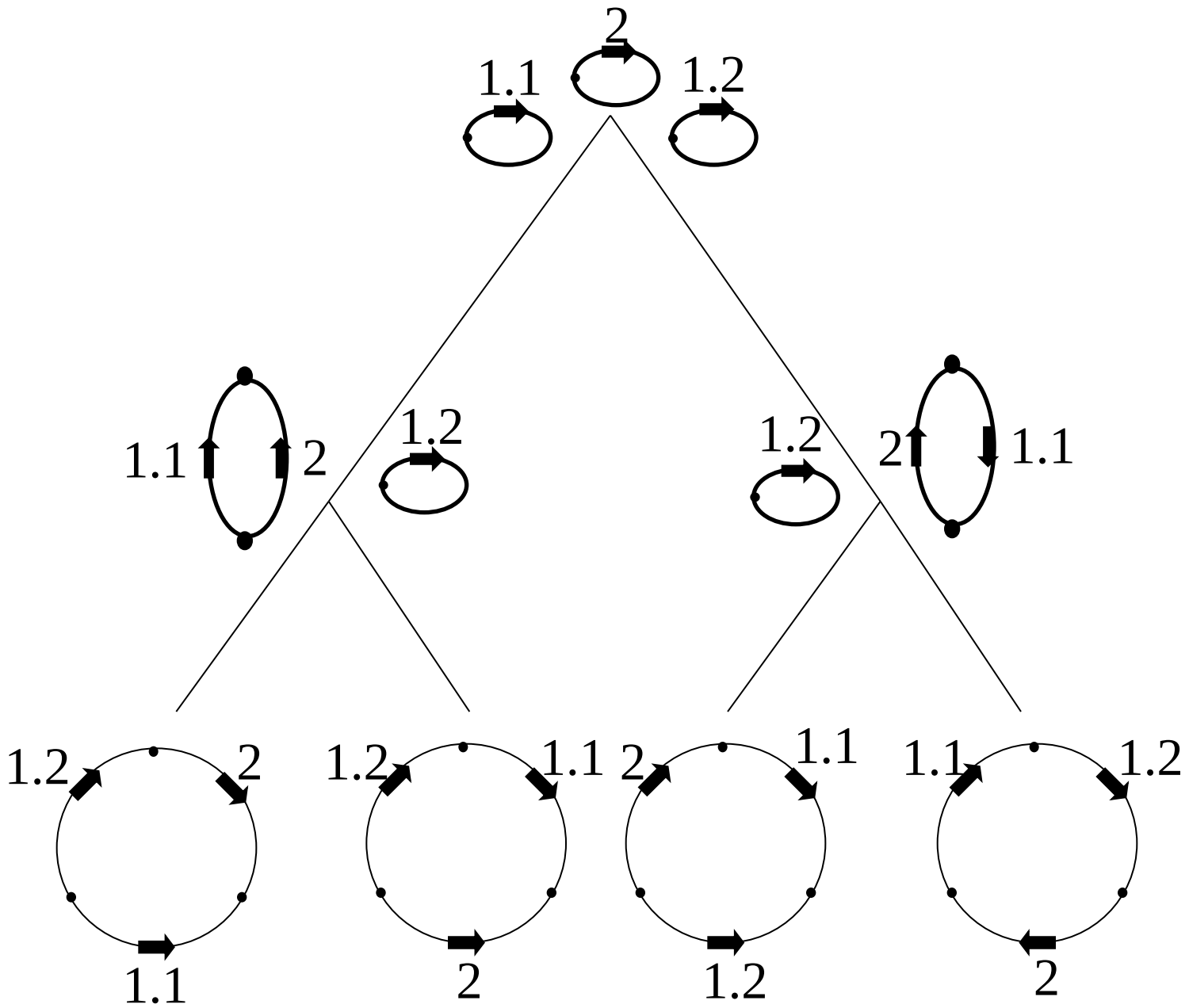


Аналогично к задаче ЦЛП сводится и **задача реконструкции** графов на данном дереве. В листьях указаны графы. Найти расстановку графов во внутренних вершинах, которая минимизирует сумму расстояний между графами по всем рёбрам дерева.

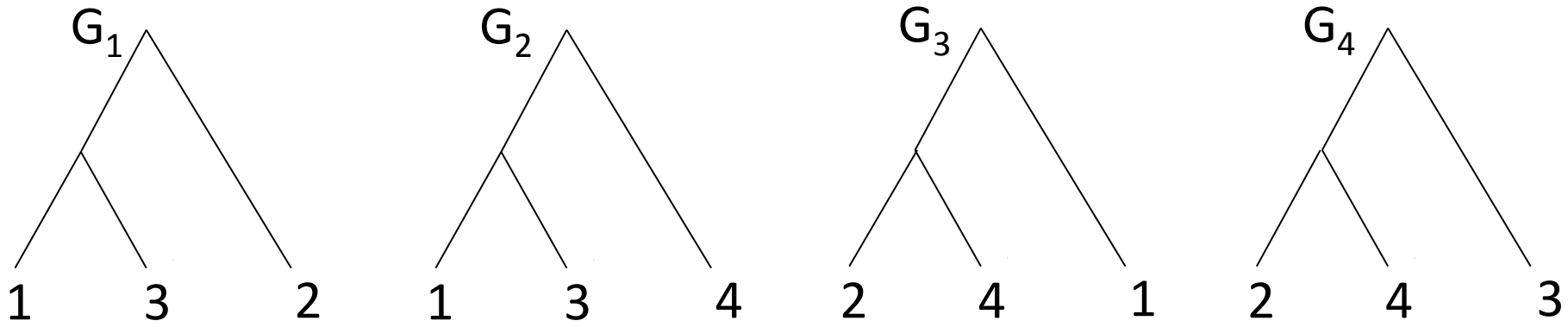
Вариант: то же самое с повторениями имён рёбер!

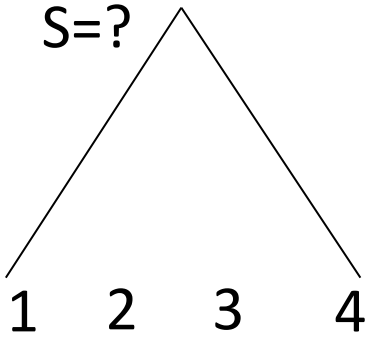


Пример: решение для данных, указанных в листьях:

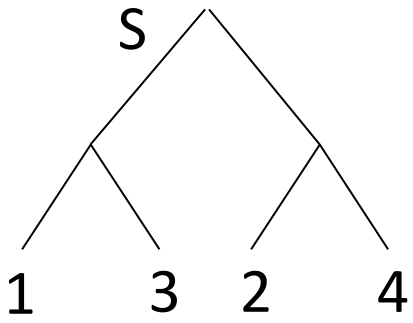


Пример:

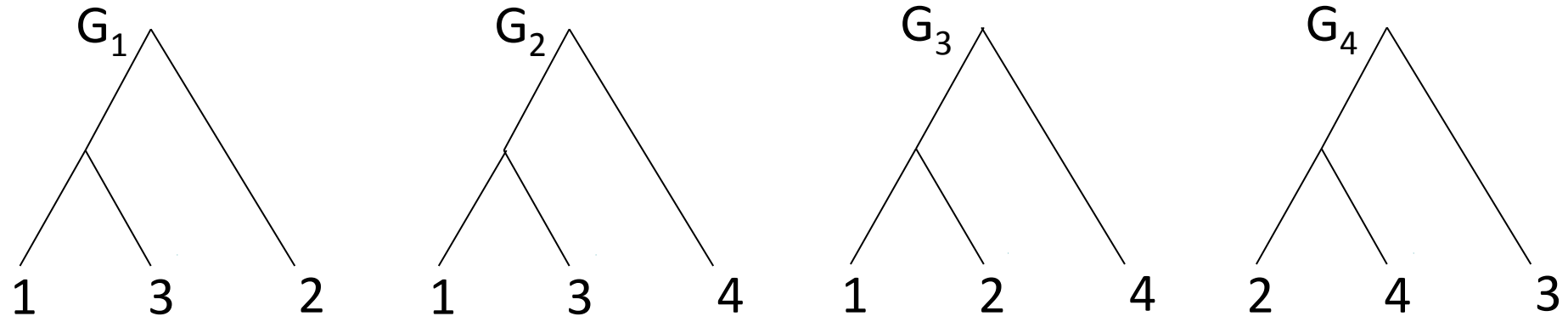


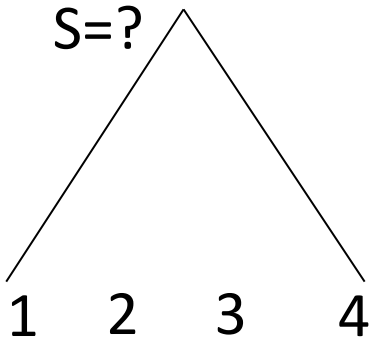


Решение:

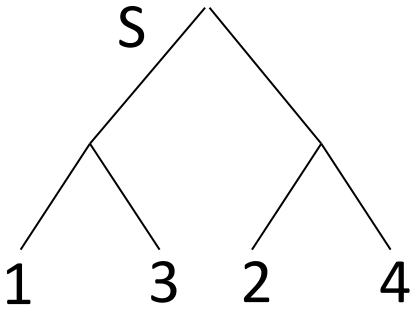


Ещё пример:

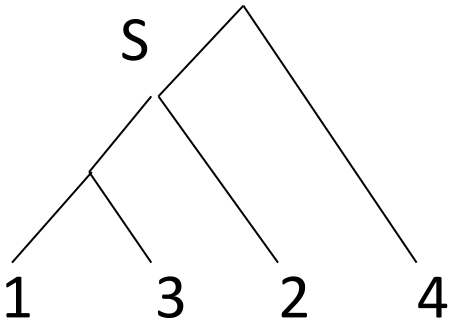




Решение:



или



Прежде всего, нужно определить **расстояние**

$$c(G_i, S)$$

от дерева G_i до дерева S . Затем **минимизируем сумму** $\sum_i c(G_i, S)$

расстояний от всех G_i до искомого S

в пространстве всех деревьев, которое пробегает переменная S . Минимизация функционала – обычное дело в прикладных задачах, но как?

Нами получен **кубический** алгоритм, который **точно!** находит такой минимум. В этом случае преимущество перед эвристическими алгоритмами велико.