Мы научились быстро находить одну строчку в другой. Но теперь мы усложняем задачу таким образом: есть две последовательности и мы хотим найти общие части у них. Про вставки, удаления мы поговорим позже.

Мы знаем BLAST и он ищет данную последовательность в большом архиве.

Есть две последовательности длины m и n, а хотим найти куски длины хотя бы k.

-Аня рассказывает алгоритм сравнивания последовательностей со сдвигом. Алена делает этот перебор удобнее начиная с момента где конец одной совпадает с началом другой. Сколько всего получится сравнений? Это наивный алгоритм сравнивает все возможные элементы, а значит ответ MN. Нарисуем таблицу M на N.

11:58

В i,j месте таблице храним 0 или 1 понятно когда. Дети думают где будут минусы и три плюса с картинки. MN мы получим, так как мы проходим по всем диагоналям последовательно. Задание – написать алгоритм реализующий наше решение.

Что делать с памятью на ответ? Её можно не учитывать.

Алена- Что делать если совпадения могут быть с реверснутой последовательностью? Аня-перевернуть и сделать тоже самое. И-надо просто пройтись по побочным диагоналям.

Как это всё ускорить? Аня и Алена - Если мы нашли не совпадение, то мы можем перепрыгнуть вперед не на 1, а на k.

12:10

И- Какие на практике k могут быть интересны? Пусть мы ищем совпадения не короче 5. вопрос к залу – сколько существует нуклеотидов длины 5? Алена – 4^5. Мы можем генерить все последовательности длины 5 и потом искать их в обеих последовательностях. И-Давайте выпишем все строки длины 5 в столбик. Взяли первую 5ку в 1последовательности, и делаем зарубочку в таблице в соотв ячейке. Вопрос на повторение: чем нам удобно воспользоваться, чтобы быстро обращаться к словарю? МА – Мы делаем лист в каждой ячейке, где ячейки образуют словарь.

12:15 Мы прошли окошком длины 5 по всему первому геному. Сколько это заняло операций?5N. Как это ускорить? МА-из 12345 мы хоrjyтим прийти в 23457. x->10\*(x/10) + new\_letter. И того посмотреть на след пятерку занимает 1 операцию. –На самом деле у нас 1 операция это не 1 битовая операция, а операция с числом небольшим. Это работает пока число не слишком большое.

12:22 Мы пробежали пока только по первой последовательности и сделали таблицу. Аня – мы идем по второй последовательности и ищем сразу это в таблице.

Сколько по времени работает алгоритм? Заметим трудность с оценкой времени, тк не ясен размер ответа. Но мы не учитываем это время.

12:27 У нас фиксированное время на нахождение строки. А суммарное время вывода пропорционален ответу. Сколько времени требуется на построение таблицы?

Итого два шага 1)построение O(m) + O(n). 2)поиск в ответе O(T) +O(A)

Что мы забыли?

12:32 Как мы строили таблицу? Инициализация таблицы стоит размера таблицы! И иногда размер таблицы больше O(mn). Но это можно улучшить. Если мы ищем большие совпадения, то мы тратим большое количество времени на построение таблицы.

Как искать 20ки? Храним 10ки. Ищем их в первой. В каждой зарубке мы ставим указатель на следующий элемент. Сколько лишних операций? порядка N. И если нашли аналогичный кусок во второй, то идем по указателю и проверяем следующую 10ку.

То что мы сделали называется хеширование.

12:43 Пусть у нас есть множество М и функция f. Мы храним не элементы М, а f(M).