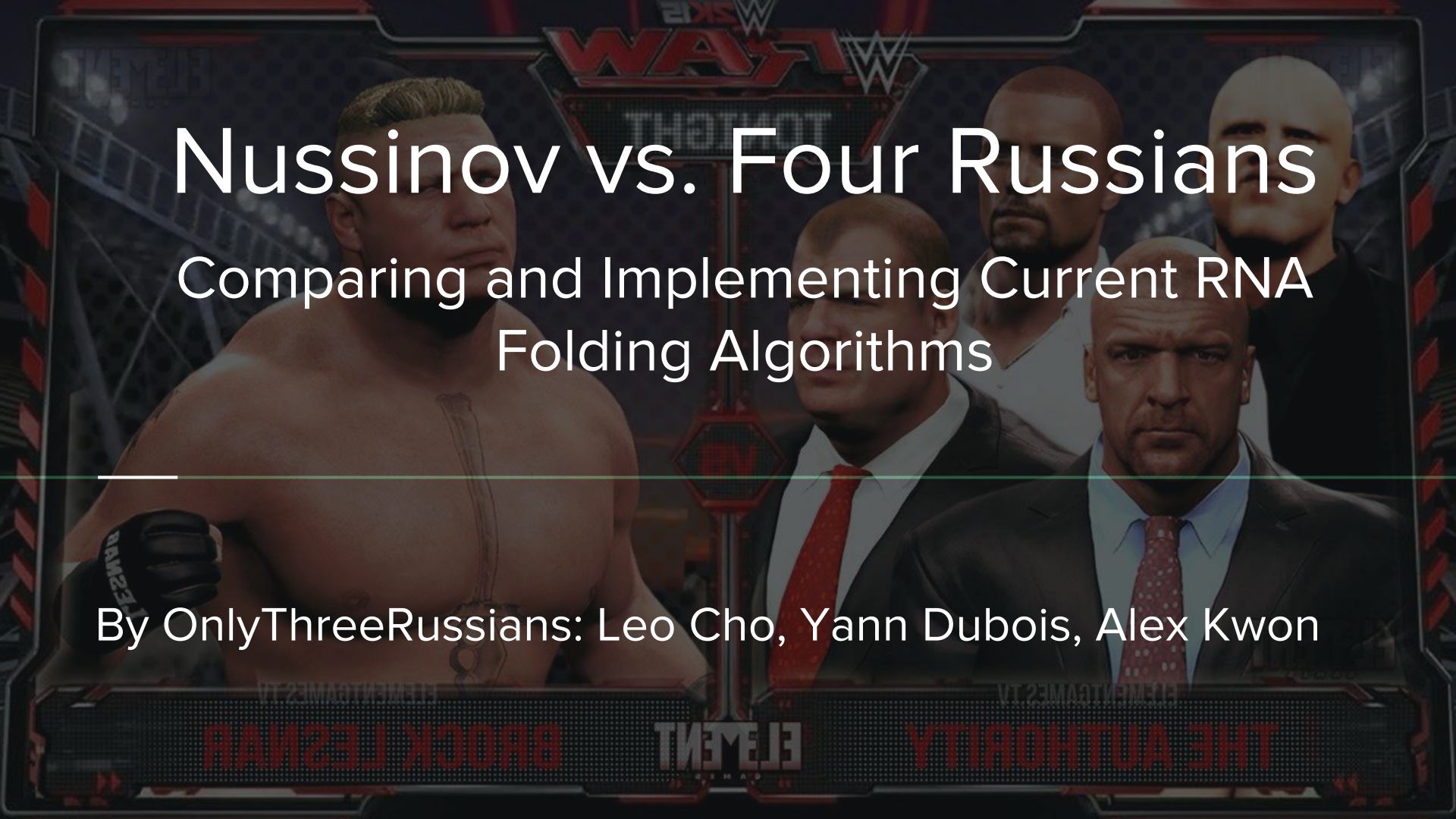
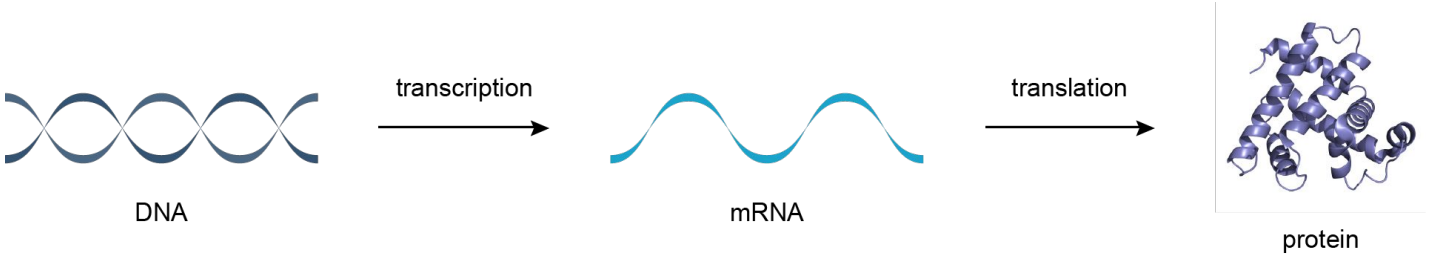


Nussinov vs. Four Russians

Comparing and Implementing Current RNA Folding Algorithms

By OnlyThreeRussians: Leo Cho, Yann Dubois, Alex Kwon

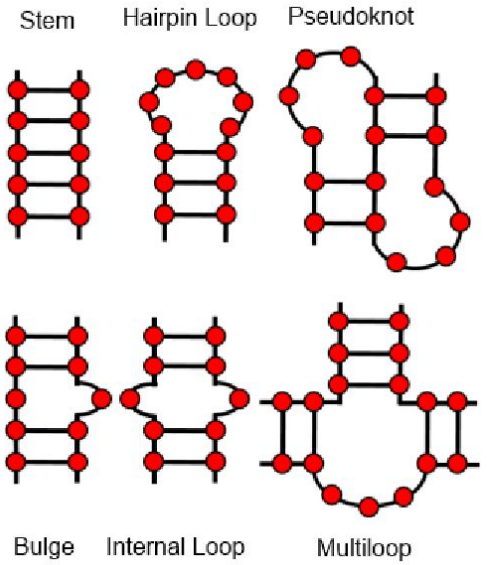
What is RNA folding?



- RNA is not just an intermediary stage between DNA and protein.
- Non-coding RNAs adopt sophisticated 3D structures and some catalyst biochemical reactions.
- Secondary structures can be conserved in homologous RNA even if the sequences are drastically different.

Secondary Structures of RNA

- Unlike DNA, RNA is produced as single stranded molecule that folds intramolecularly
- Base pairs almost always occurs in a nested fashion in secondary structures (Non-crossing matches)
- If *i* and *j* are a pair and *k* and *l* are a pair, then the positions of the bases are either:
  - $i < j < k < l$ (non-intersecting)
  - $i < k < l < j$ (nested)
  - else : pseudoknot
- Using dynamic programming, calculate probable secondary structures by optimizing for minimizing free energy.



The Goal:
Implement and compare Nussinov and Four Russians algorithm against 2kbp, 3kbp, 4kbp, 5kbp, and 6kbp RNA sequences

In the blue corner..... Nussinov

Nussinov's algorithm

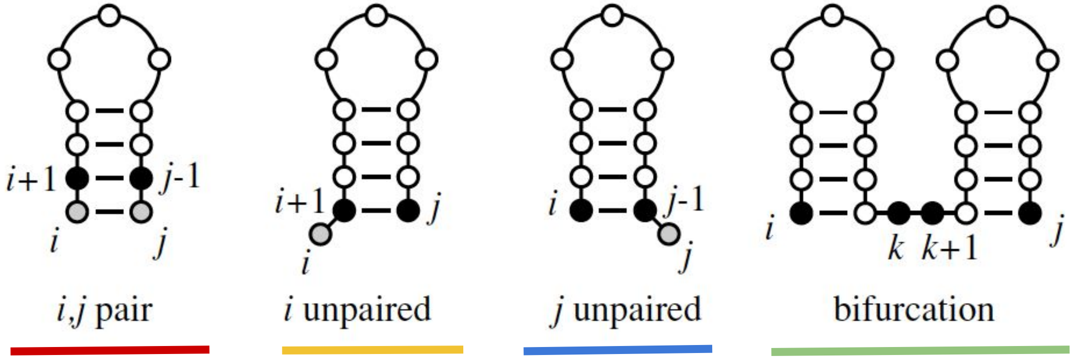
- Maximize the number of **non-crossing** complementary base pair matches
- Cannot match with base just before
- Let ***D(i,j)*** be optimal cost of folding for subsequence *i* to *j*
- DP table built column-wise or diagonally
- **O(n^3)** runtime

- Initialisation:

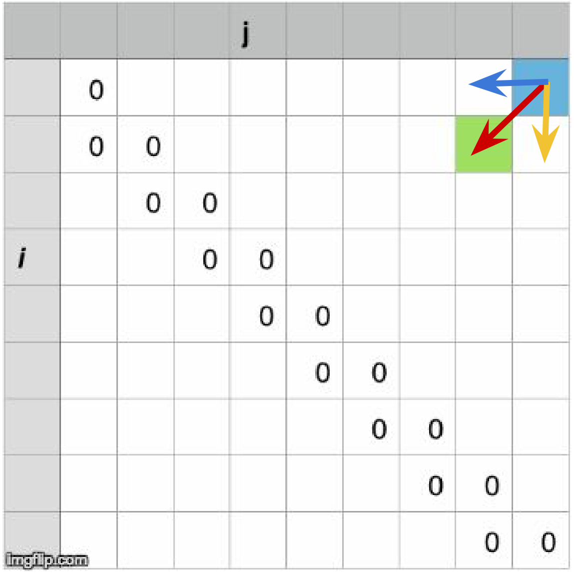
$$D(i,j) = \max \left\{ \begin{array}{cc} D(i,i) = 0 & \forall i = 1..L \\ D(i,i-1) = 0 & \forall i = 2..L \end{array} \right\}$$

- Termination: Optimal solution given by *D(1, n)* then traceback

Recurrence



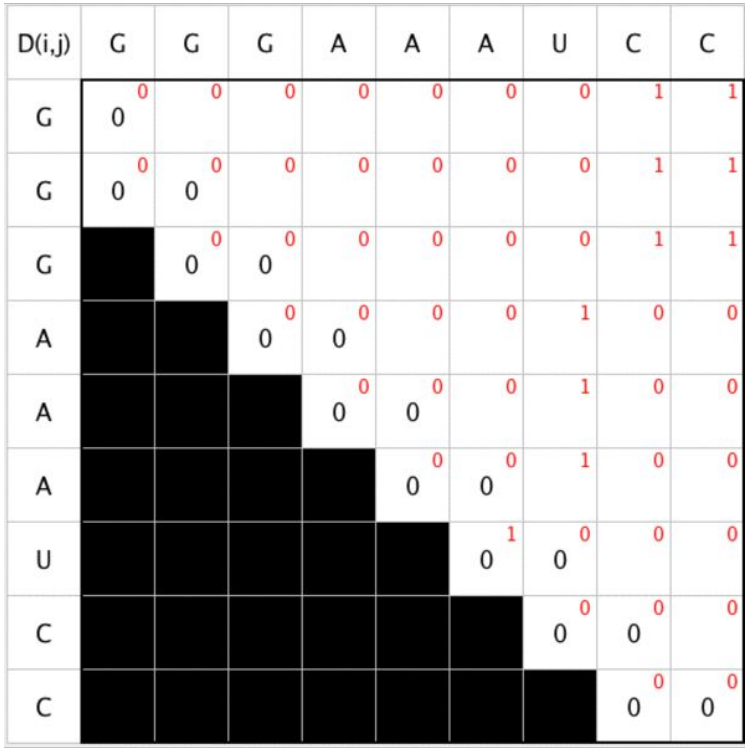
$$D(i,j) = \max \begin{cases} max_{i<k<j} D(i,k) + D(k+1,j) \\ D(i+1,j-1) + \ w(i,j) \\ D(i+1,j) \\ D(i,j-1) \end{cases}$$



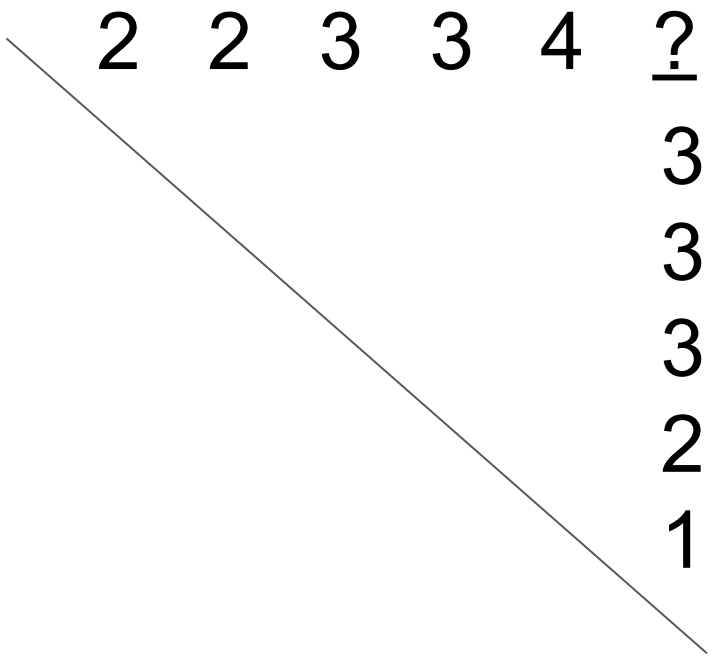
Nussinov’s example and complexity

- Space complexity : **O(n^2)**

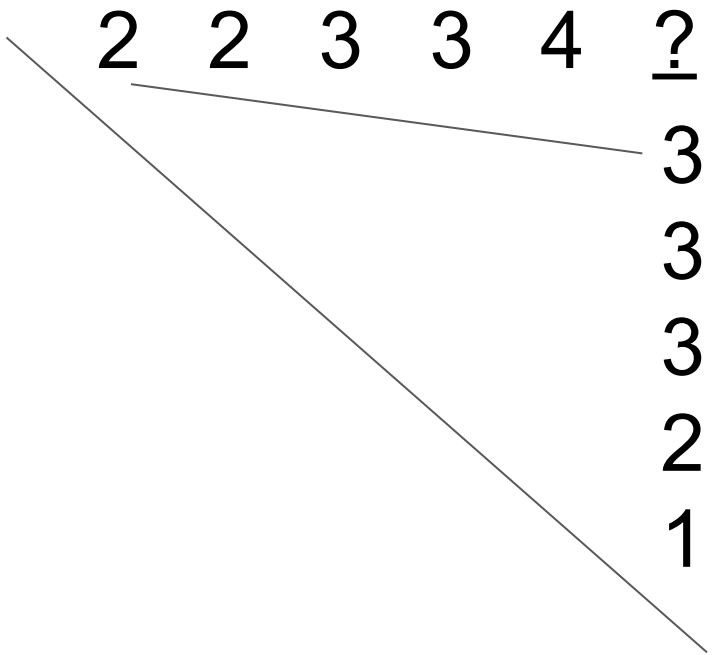
- Time complexity: **O(n^3)**
- For each O(n^2) cells have to look at all O(n) previous cells in the same line / column



In the red corner……. the Four Russians

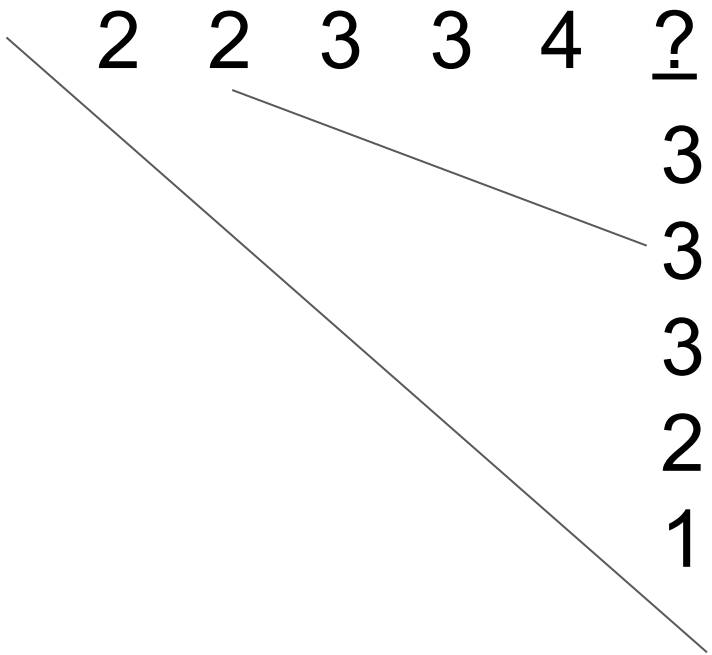


$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$



$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

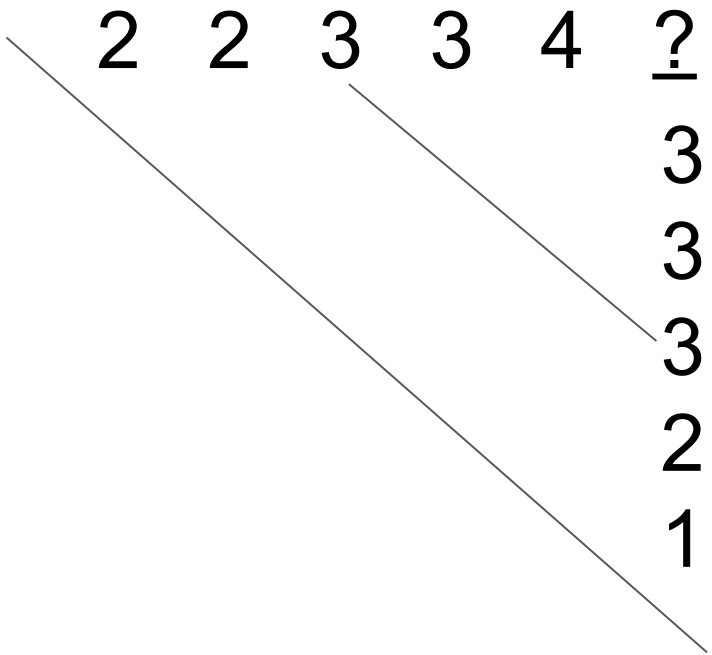
2 + 3 = 5



$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

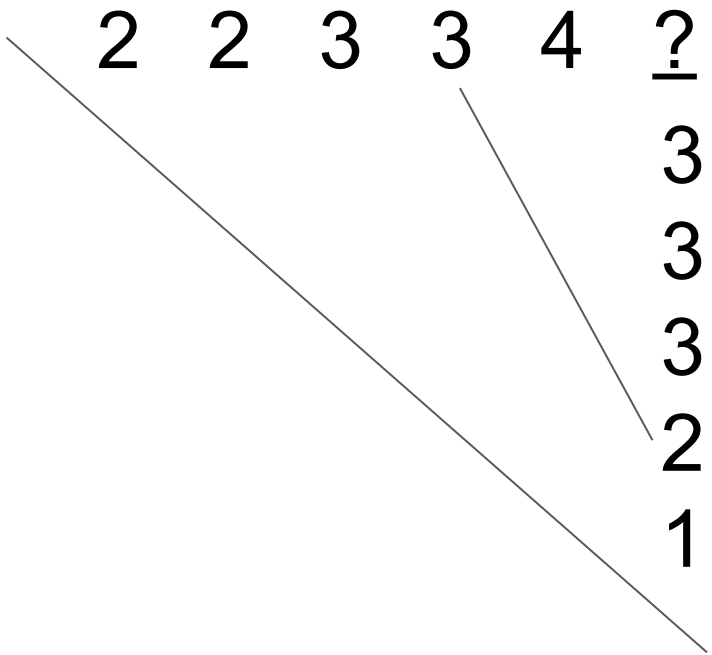
2 + 3 = 5

2 + 3 = 5



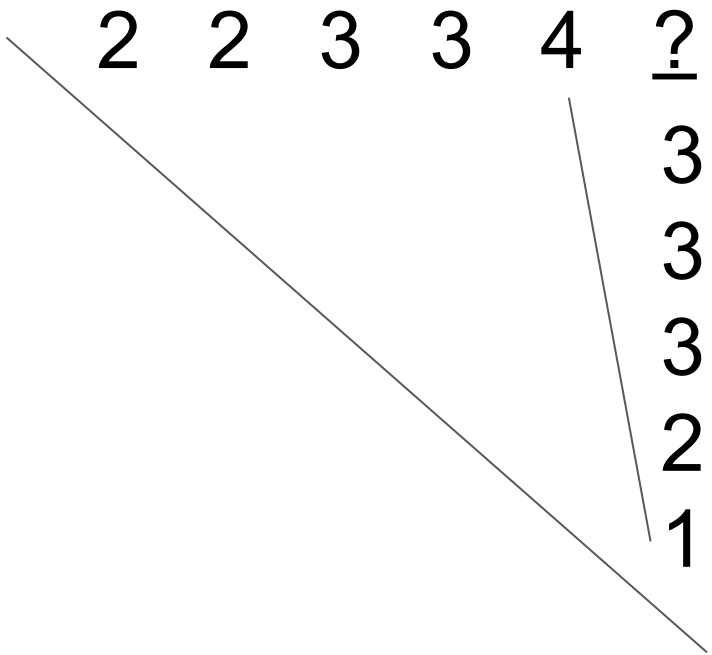
$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

2 + 3 = 5
2 + 3 = 5
3 + 3 = 6



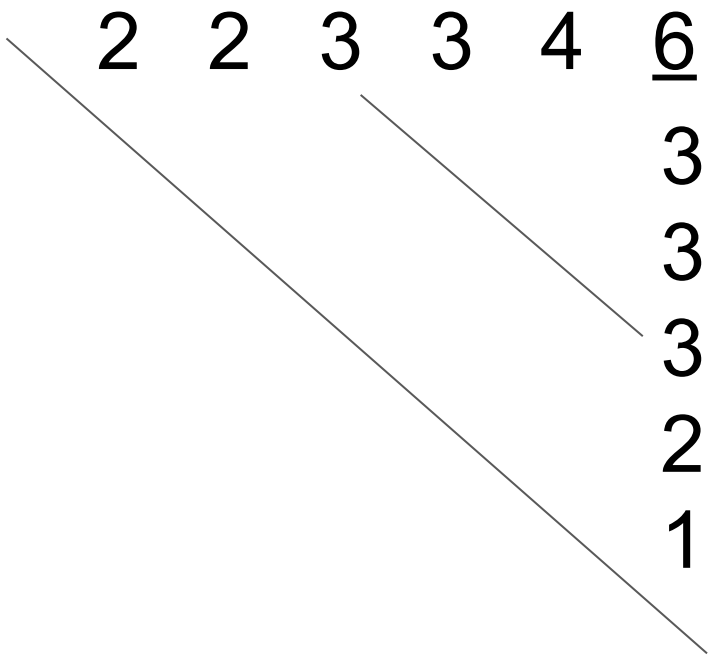
$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

2 + 3 = 5
2 + 3 = 5
3 + 3 = 6
3 + 2 = 5



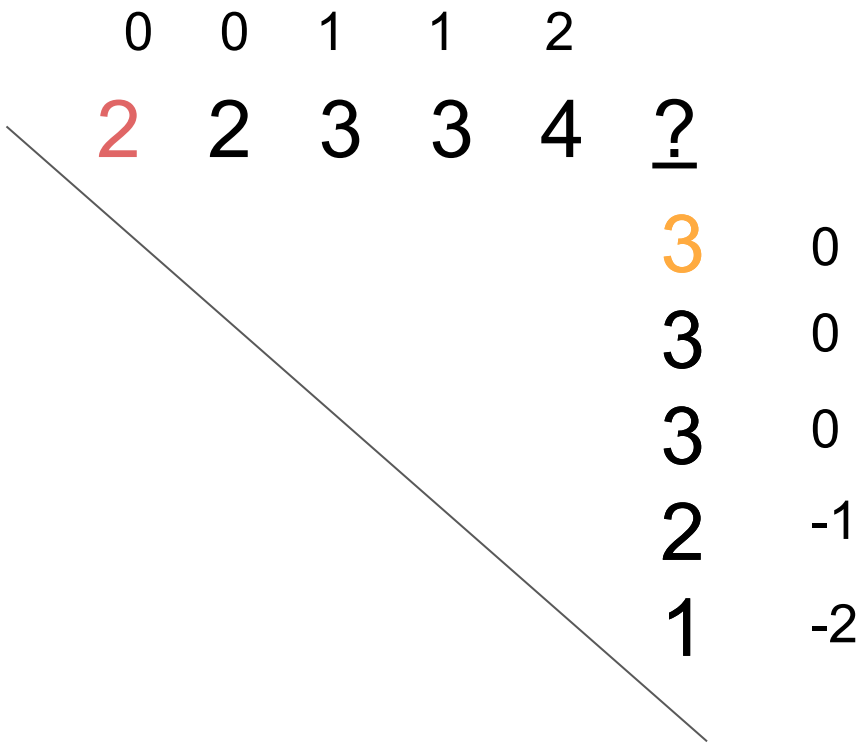
$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

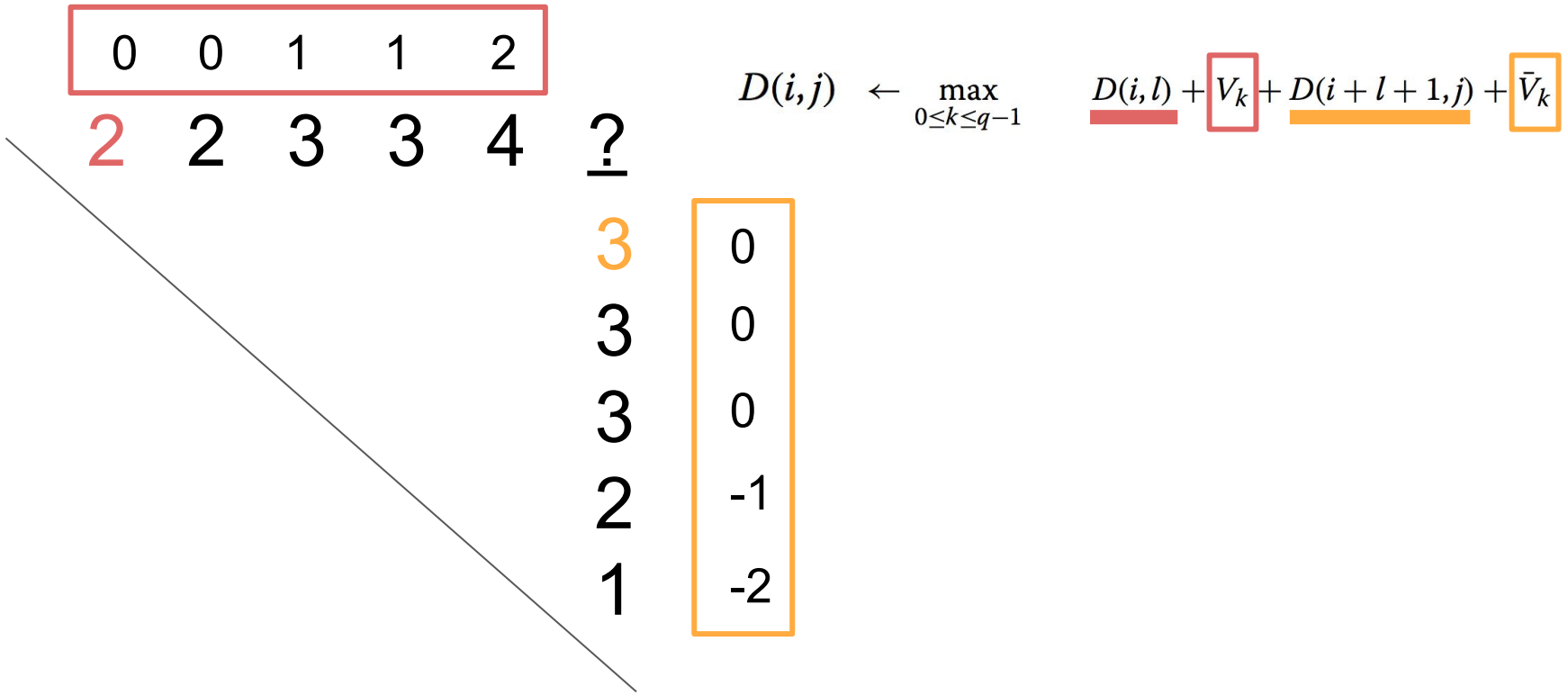
2 + 3 = 5
2 + 3 = 5
3 + 3 = 6
3 + 2 = 5
4 + 1 = 5

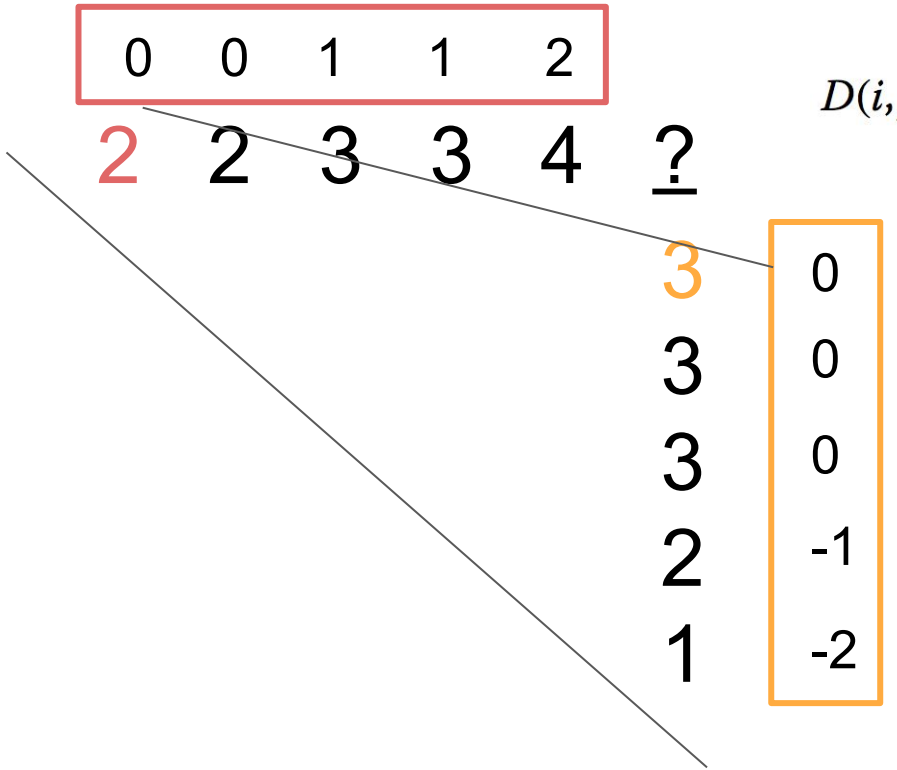


$$D(i,j) \leftarrow \max_{l+1 \le k \le l+q} D(i,k-1) + D(k,j)$$

2 + 3 = 5
2 + 3 = 5
3 + 3 = 6
3 + 2 = 5
4 + 1 = 5

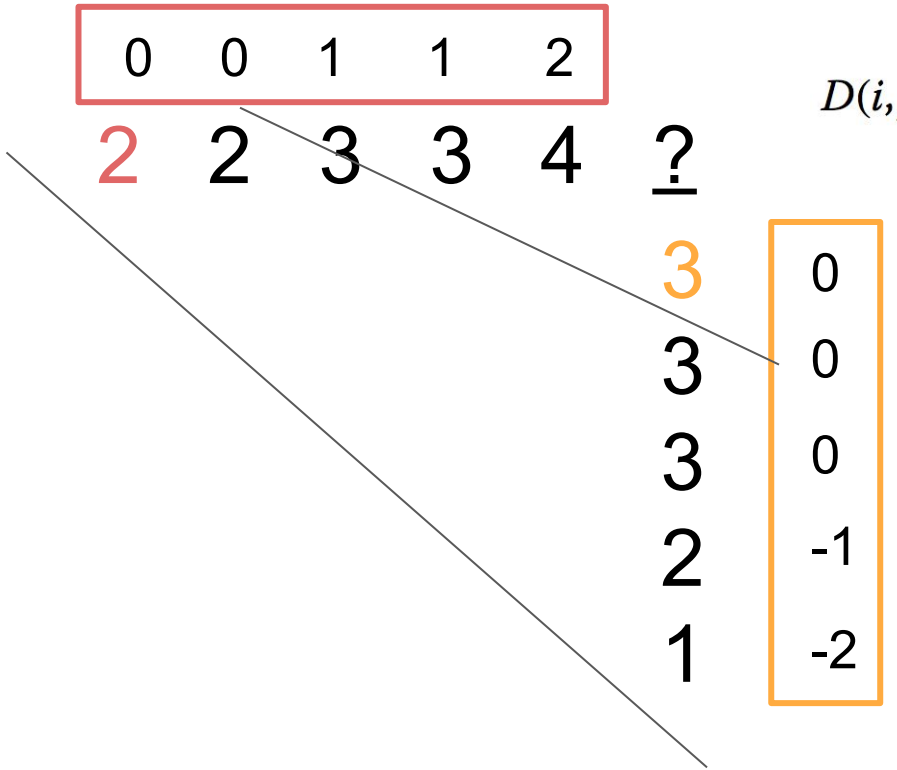

0 0 1 1 2
2 2 3 3 4 ?
3 0
3 0
3 0
2 -1
1 -2


0 0 1 1 2
2 2 3 3 4 ?
3 0
3 0
3 0
2 -1
1 -2
$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$



$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

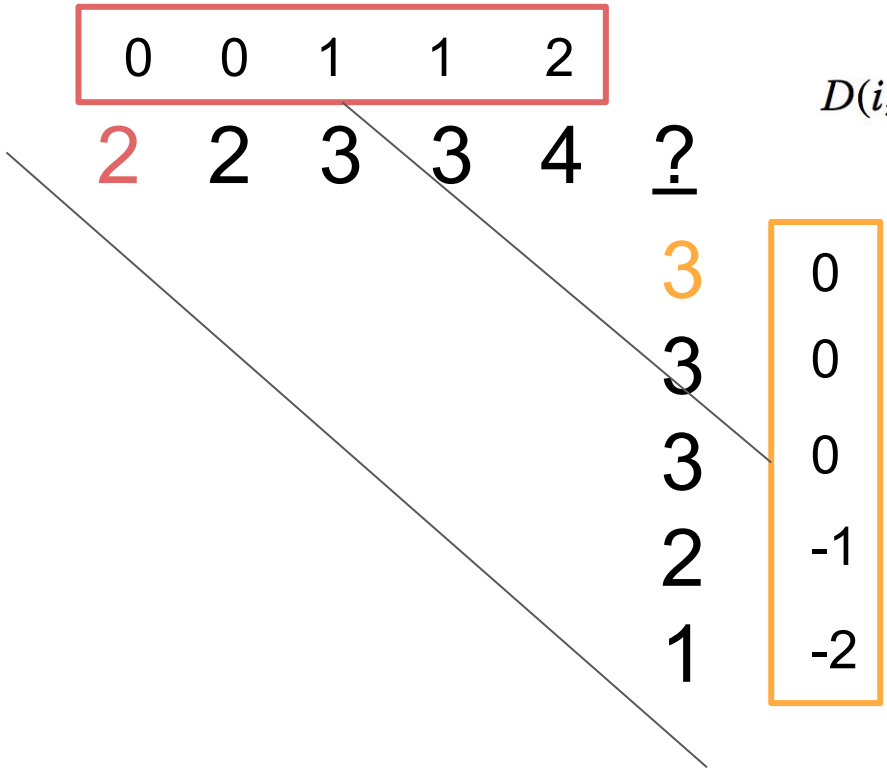
2 + 0 + 3 + 0 = 5



$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

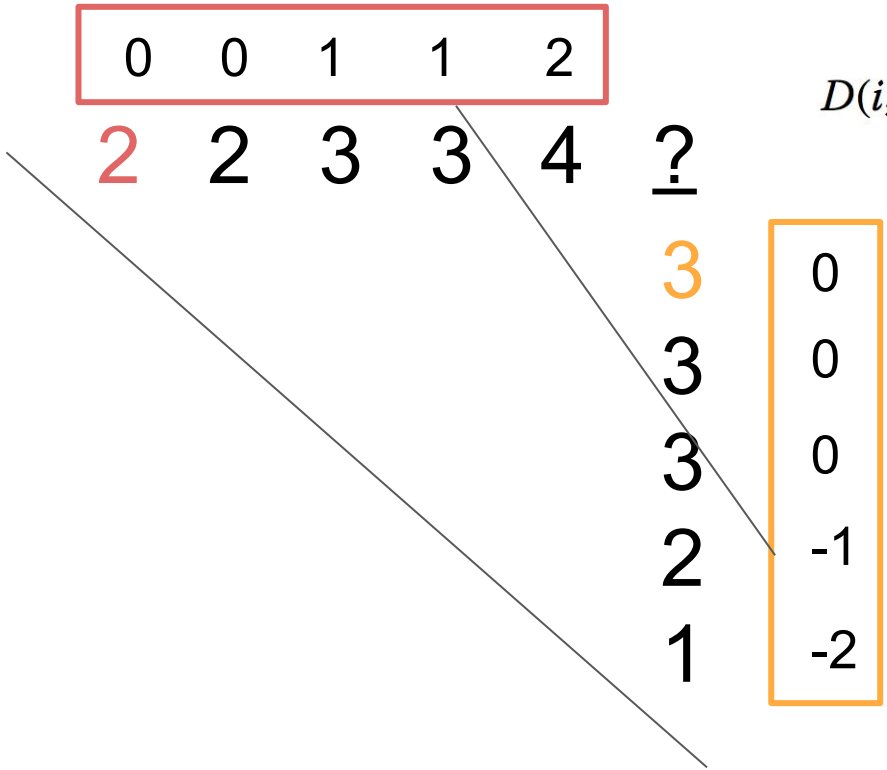
2 + 0 + 3 + 0 = 5

2 + 0 + 3 + 0 = 5



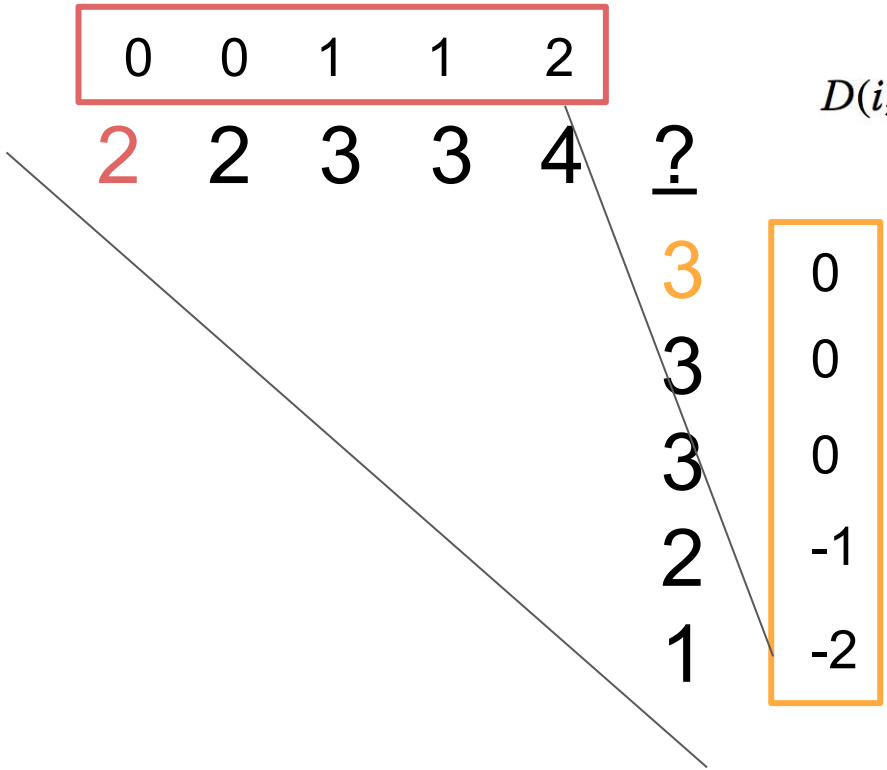
$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

2 + 0 + 3 + 0 = 5
2 + 0 + 3 + 0 = 5
2 + 1 + 3 + 0 = 6



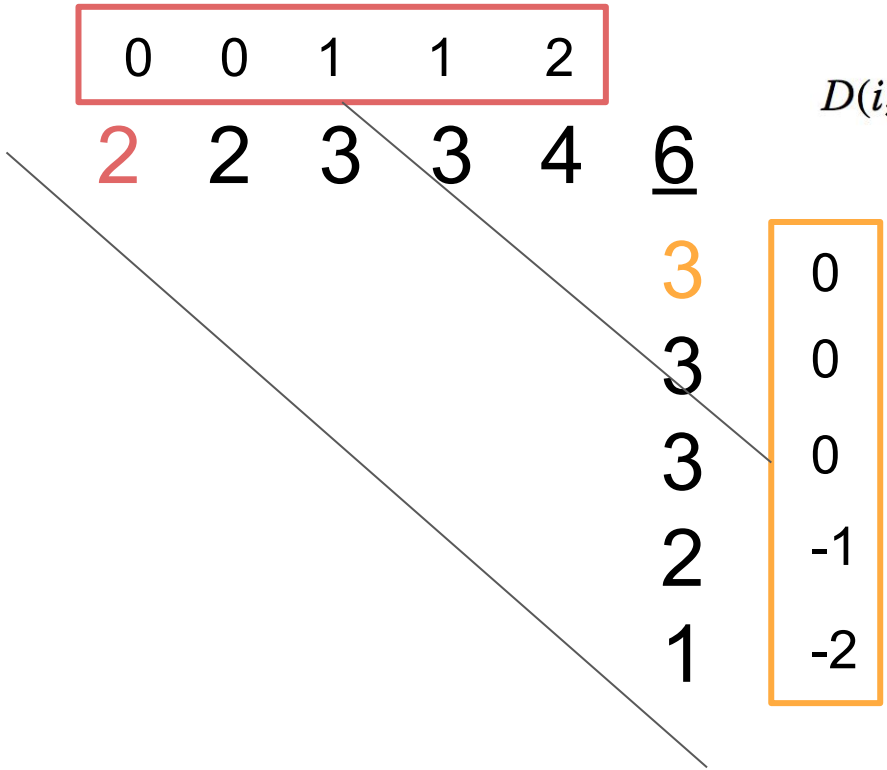
$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

2 + 0 + 3 + 0 = 5
2 + 0 + 3 + 0 = 5
2 + 1 + 3 + 0 = 6
2 + 1 + 3 + (-1) = 5



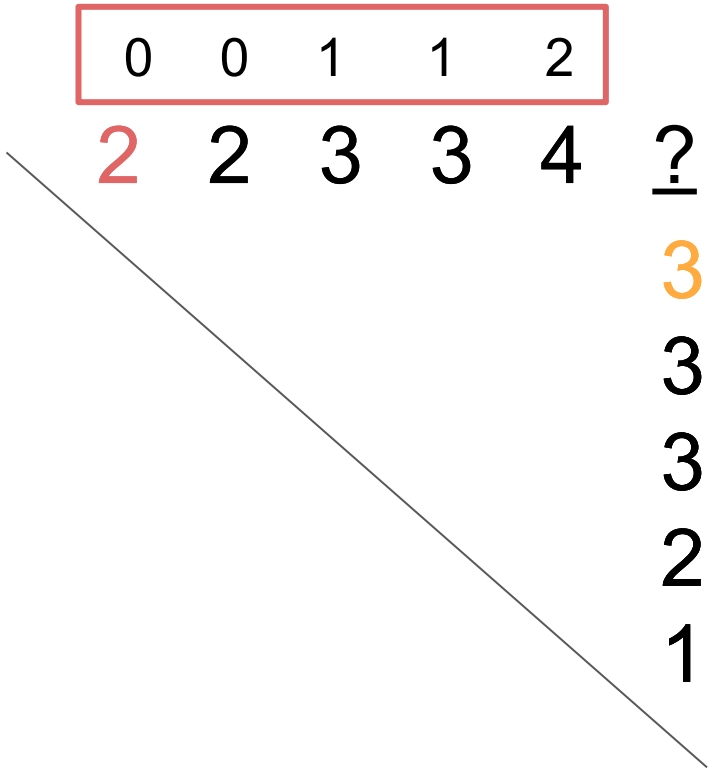
$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

2 + 0 + 3 + 0 = 5
2 + 0 + 3 + 0 = 5
2 + 1 + 3 + 0 = 6
2 + 1 + 3 + (-1) = 5
2 + 2 + 3 + (-2) = 5



$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

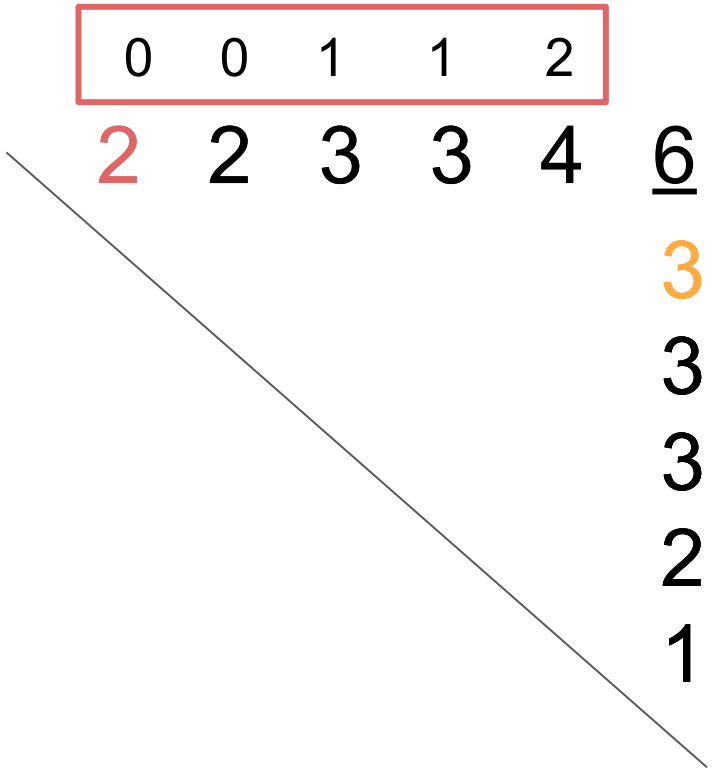
2 + 0 + 3 + 0 = 5
2 + 0 + 3 + 0 = 5
2 + 1 + 3 + 0 = 6
2 + 1 + 3 + (-1) = 5
2 + 2 + 3 + (-2) = 5



$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

$$\leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + max (
0 + 0 = 0
0 + 0 = 0
1 + 0 = 1
1 + (-1) = 0
2 + (-2) = 0)

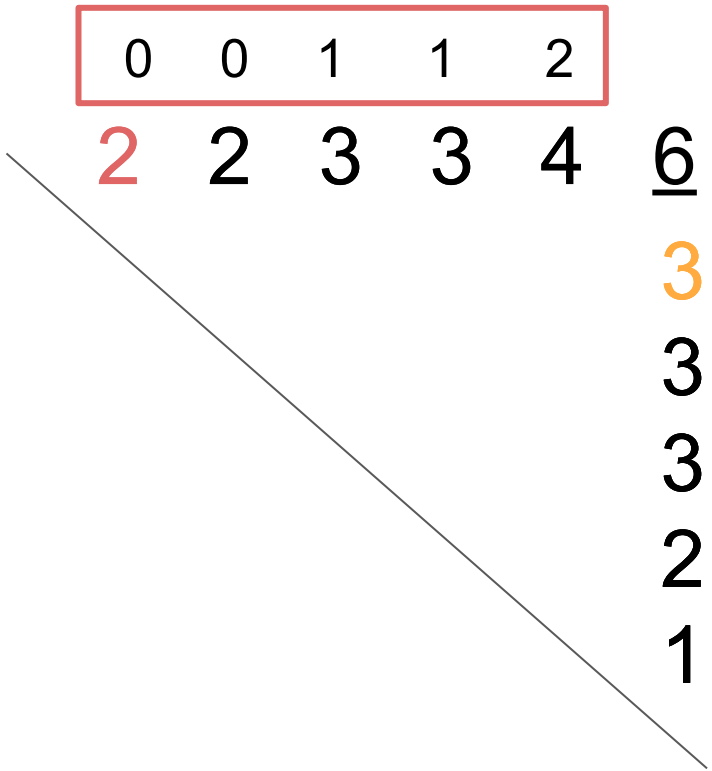


$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

$$\leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + max (
0 + 0 = 0
0 + 0 = 0
1 + 0 = 1
1 + (-1) = 0
2 + (-2) = 0)

2 + 3 + 1 = 6

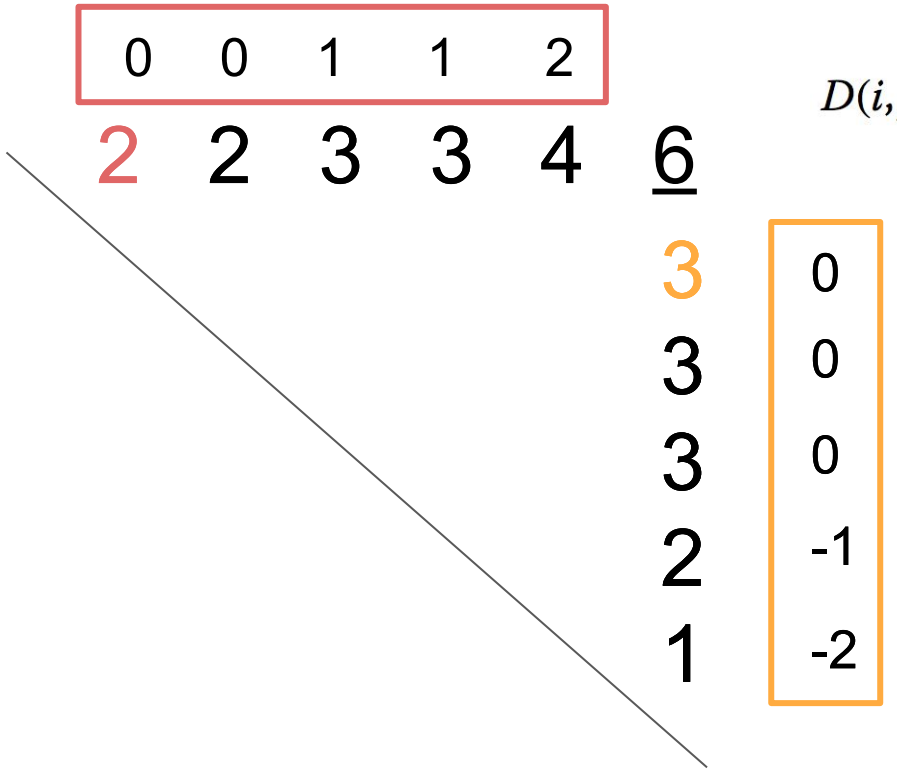


$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

$$\leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + maxDiff ([0,0,1,1,2], [0,0,0,-1,-2])

2 + 3 + 1 = 6



$$D(i,j) \leftarrow \max_{0 \le k \le q-1} D(i,l) + V_k + D(i+l+1,j) + \bar{V}_k$$

$$\leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + maxDiff ([0,0,1,1,2], [0,0,0,-1,-2])

2 + 3 + 1 = 6

These running total vectors have q! permutations.

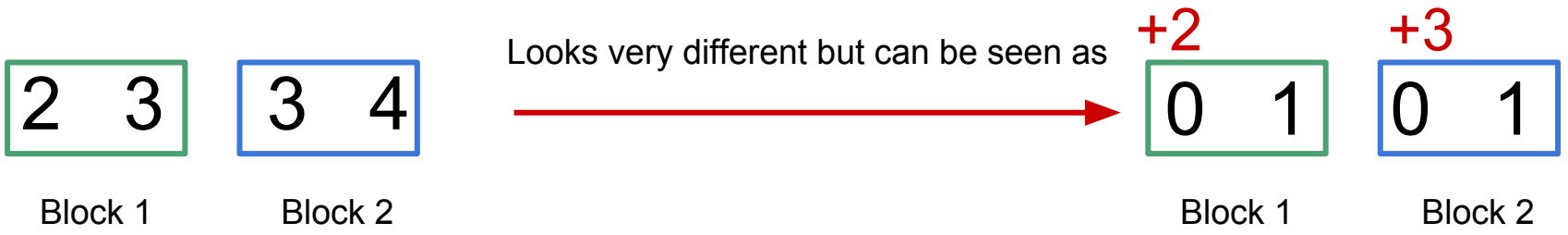
maxDiff is too large to precompute.

Speed-up intuition

- Intitial idea:

Subset into blocks and precompute best subalignement from blocks.

Note: useless because values of blocks would be very different so rarely reuses computation.



Speed-up intuition

- Lemma:

Values along column (bottom -> top) and row (left -> right) are non decreasing and increasing from at most 1. Because added at most a single match.

- Obsevation:

If we precompute the values based on subsets of columns / rows we would be able to reuse computation.

horizontal difference vector

0	1	0	1
0	0	1	1	2

vertical difference vector

0	0	1	1
0	0	0	-1	-2

$$D(i,j) \leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + maxDiff ([0,0,1,1,2], [0,0,0,-1,-2])

2 + 3 + 1 = 6

horizontal difference vector

0	1	0	1

0	0	1	1	2

vertical difference vector

0	0	1	1

0	0	0	-1	-2

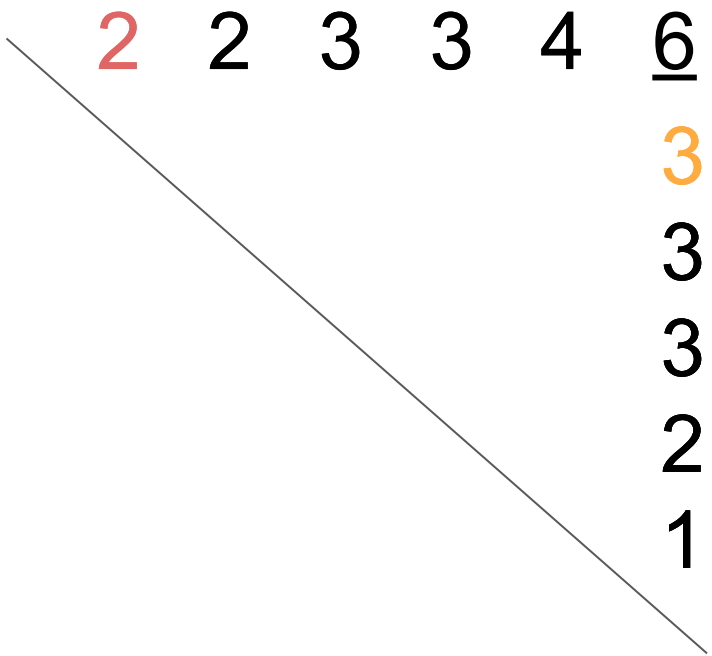
$$D(i,j) \leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

2 + 3 + maxDiff (**[0,1,0,1], [0,0,1,1]**)

2 + 3 + 1 = 6

difference vectors
have 2^q permutations.

Feasible to precompute maxDiff for reasonably sized q.



D(i,j) now computable in constant time for q consecutive cells.

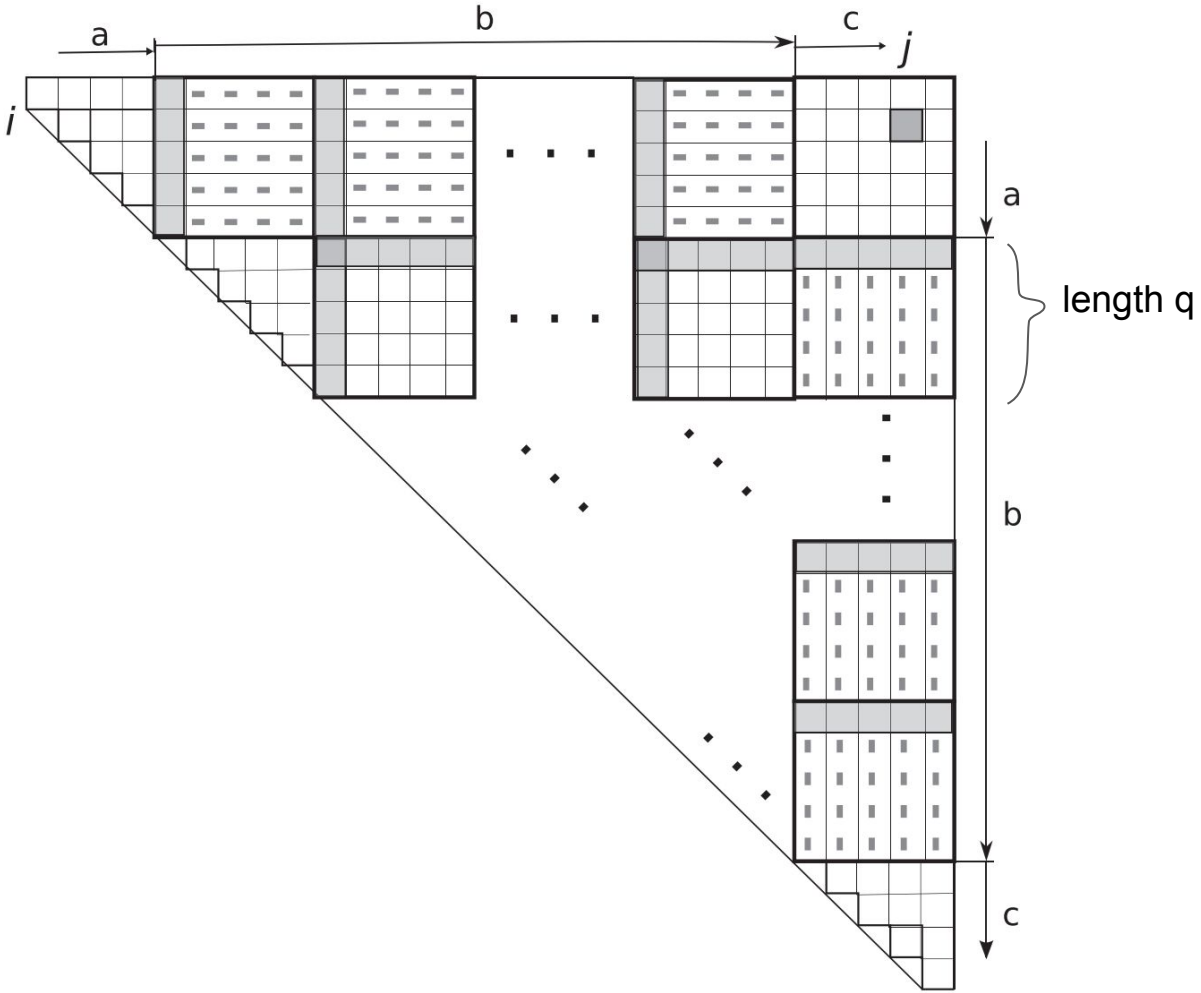
$$D(i,j) \leftarrow D(i,l) + D(i+l+1,j) + \max_{0 \le k \le q-1} V_k + \bar{V}_k$$

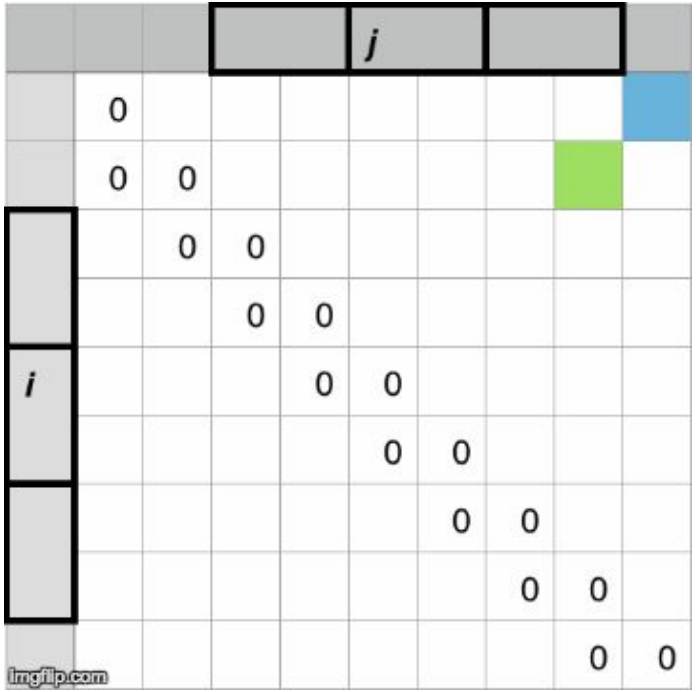
2 + 3 + maxDiff (**[0,1,0,1], [0,0,1,1]**)

2 + 3 + 1 = 6

difference vectors
have 2^q permutations.

Feasible to precompute maxDiff for reasonably sized q.


a
b
c
j
i
a
length q
b
c


j
i
0
imgflip.com

Complexity

- Preprocessing:

Q-length vectors are binary so 2^q possible such vectors.

I.e 2^(2*q) pairs of vertical/horizontal vectors.

Computation per vector is O(q). so **O(q2^(2q)).**

Let q = log_2(n)

so O(log(n)*2^(2log(n))) = O(log(n)*2^(log(n)^2))= **O(n^2 log(n))**

Complexity

- Iterations:

O(n^2) cells to fill.

Each cell need to look at O(n/q) groups

Accessing preprocessed function is O(1) so “computing” score of group is O(1).

q = log(n)

O(n^2 * n/q) = O(n^2 * n/log(n)) = **O(n^3/log(n))**

FIGHT!

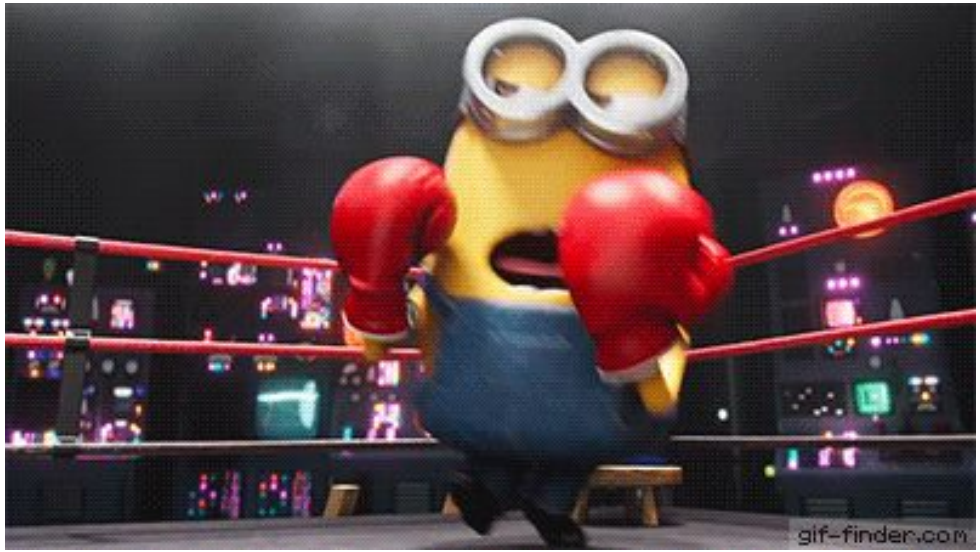
ROUND 1: **2094 bp** NPTN

ROUND 2: **2704 bp** NOP2 nucleolar protein

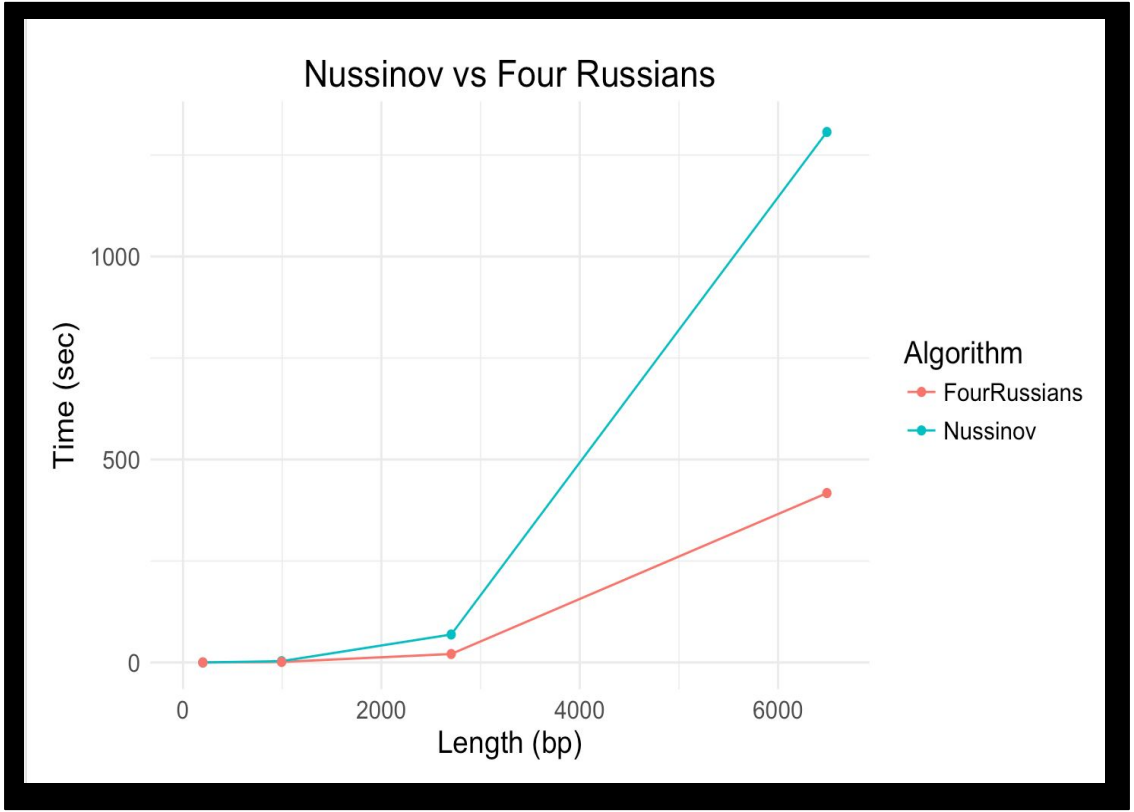
ROUND 3: **3800 bp** BRCA1

ROUND 4: **4900 bp** CD28

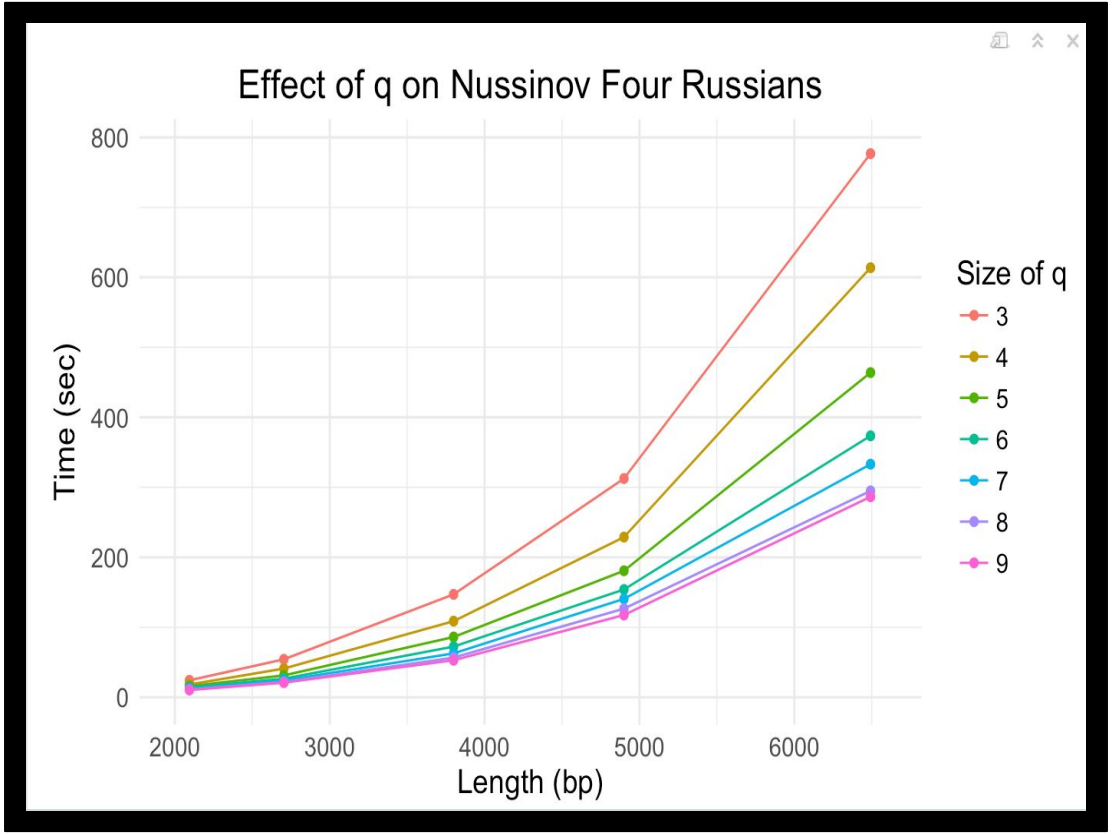
ROUND 5: **6492 bp** BCL2



Results


Nussinov vs Four Russians
Time (sec)
1000
500
0
0
2000
4000
6000
Length (bp)
Algorithm
FourRussians
Nussinov

Results


Effect of q on Nussinov Four Russians
Time (sec)
800
600
400
200
0
2000
3000
4000
5000
6000
Length (bp)
Size of q
3
4
5
6
7
8
9

Results

Size of *q*	NPTN (2094 bp)	NOP2 (2704bp)	BRCA1 (3800bp)	CD28 (4900bp)	BCL2 mRNA (6492bp)
3	24s 251ms	54s 266ms	2m 27s 78ms	5m 12s 529ms	12m 56s 741ms
4	18s 277ms	41s 83ms	1m 48s 792ms	3m 48s 878ms	10m 13s 784ms
5	15s 890ms	31s 285ms	1m 26s 175ms	3m 0s 797ms	7m 43s 865ms
6	14s 62ms	26s 593ms	1m 12s 390ms	2m 34s 39ms	6m 13s 604ms
7	12s 259ms	24s 92ms	1m 2s 765ms	2m 20s 607ms	5m 33s 85ms
8	10s 997ms	21s 691ms	56s 710ms	2m 6s 655ms	4m 54s 885ms
9	10s 510ms	20s 972ms	52s 985ms	1m 57s 582ms	4m 46s 541ms
Score	858	1093	1557	2029	2688

Algorithm	seq (994 bp)	BRCA1 (3800bp)	BCL2 mRNA (6492bp)
Nussinov	3s 237ms	1m 8s 89ms	21m 46s 760ms
FourRussians	1s 327ms	20s 861ms	6m 57s 328ms
Score	392	1557	2688

Conclusion

- The Four Russians does speed up the Nussinov algorithm for RNA folding
- The greater the size of *q*, faster speed up factor
- The biggest downside to these particular algorithms is that they cannot handle crossing matches thus cannot represent pseudoknot secondary structures (although pseudoknots make up only a small fraction of observed structures)
- There are more accurate parameters than maximum non-crossing base pairs:
    - Thermodynamics
    - Probabilistic models
    - Combination of thermodynamics and statistical parameters
- The algorithm can be further improved by speeding up Nussinov's algorithm by a factor of log^2(n) - Y.Song (2015)

Research

Faster algorithms for RNA-folding using the four-russians method by B. Venkatachalam, D. Gusfield, and Y. Frid

Time and Space Efficient Algorithms for RNA Folding with the Four-Russians Technique by Y. Song

Code

https://github.com/YannDubs/FourRussiansRNA