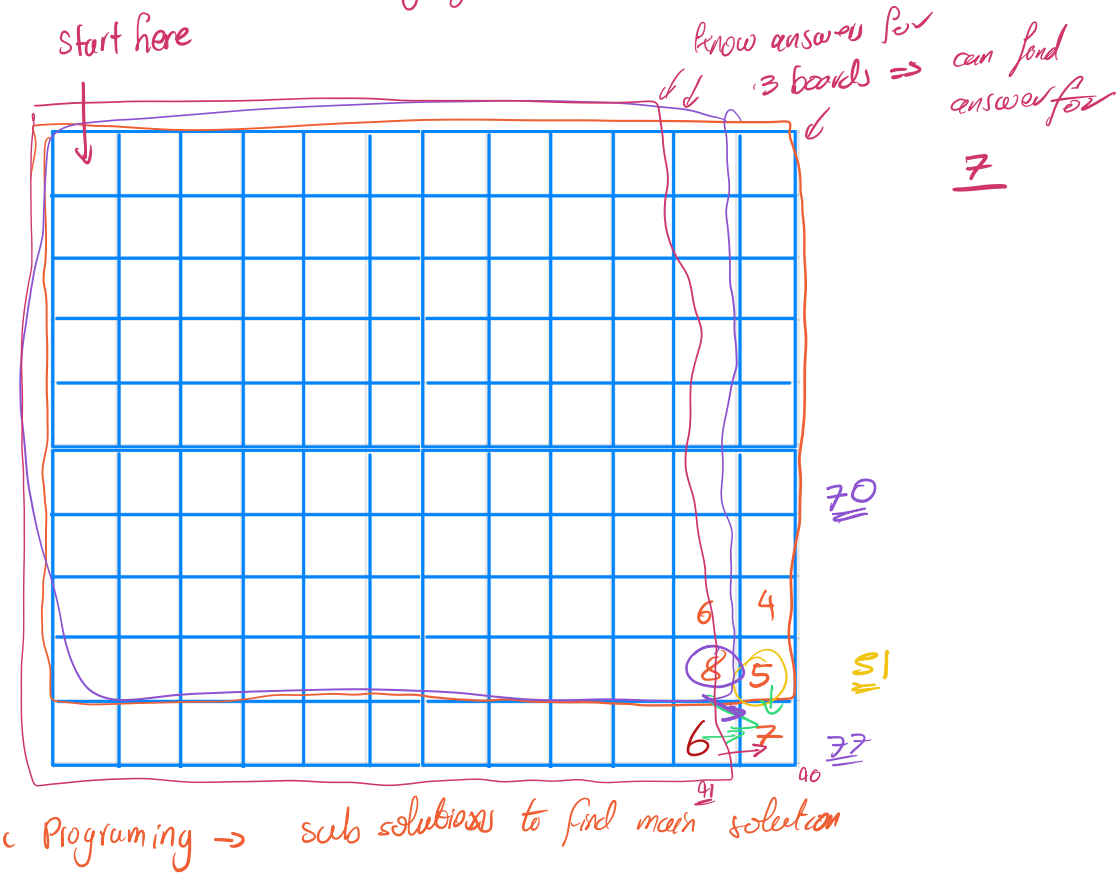
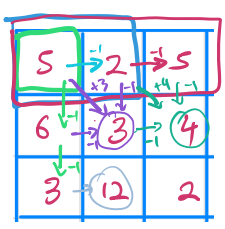
Total money you can make?

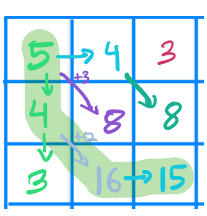


Dynamic Programming → sub solutions to find main solution



Game Board

Now, backtrack to find best path
=> recursive



Dynamic Prog Table
→ track highest money for any score

Memory

how much additional memory to store numbers?

Time

max money for each square

Computational Complexity → abstraction to quantify time algorithm takes

↳ more complex algorithm → ↑ time complexity

↓

quantifies how long it takes

↓

form of equation → e.g. linear, quadratic etc

Complexity:

Big "O" notation

$O(k)$ ← linear coefficient

↑ algorithm ⇒ $O(r \cdot c)$ — rows × colums

Time complexity is linear to $r \cdot c$

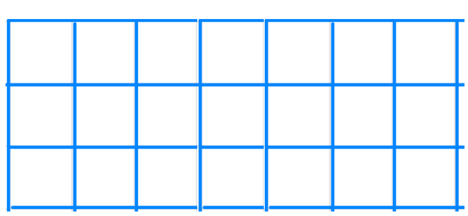
Memory Complexity (space)

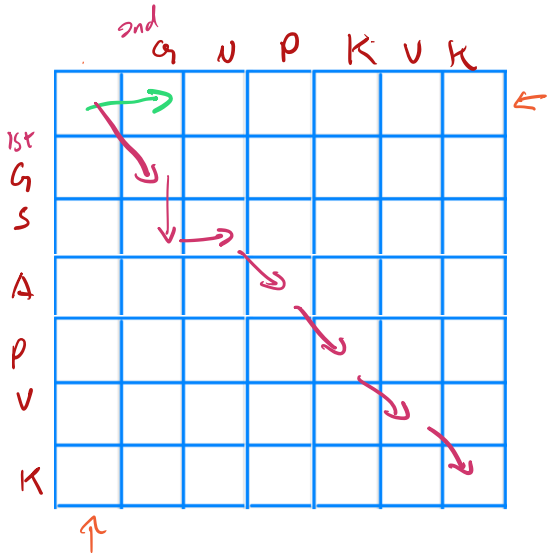
Time & Memory requirements → depend on algorithm

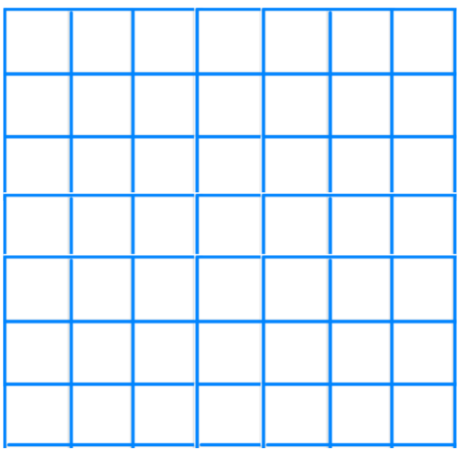
- all data stored ⇒ $O(r \cdot c)$ → On^2
- rows deleted, colums kept ⇒ $O(c)$ → On
- colums kept, rows deleted ⇒ $O(r)$ → On

let $r = n$, $c = n$

Game Board for Sequence Alignment





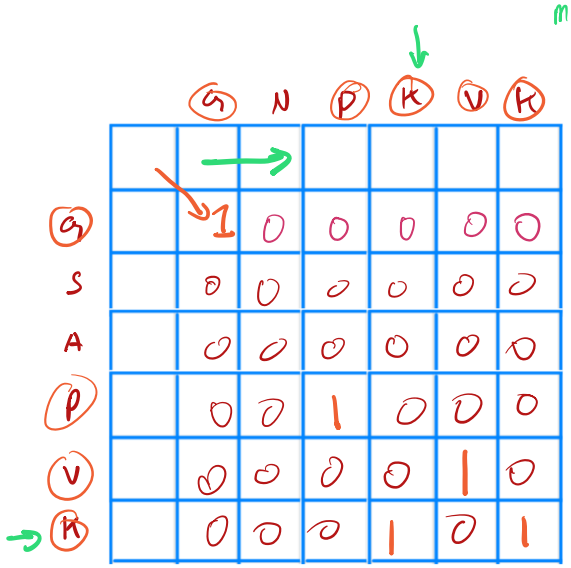


empty columns allow gaps

G S - A P V K
G - N P K V K

→ ↓ = gaps
↘ = match/mismatch

↓ => gap in second
→ => gap in first



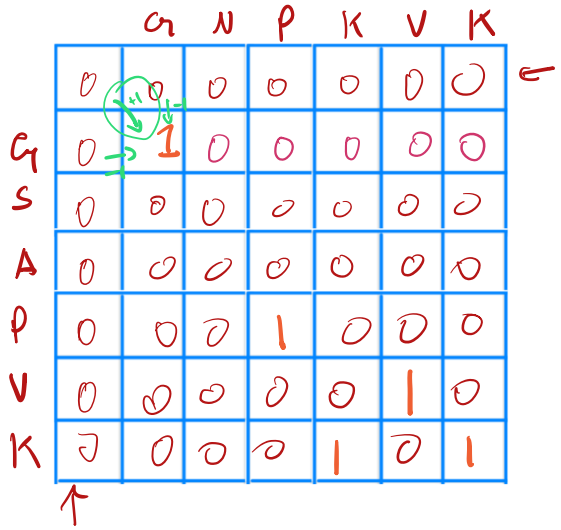
matching there => +1

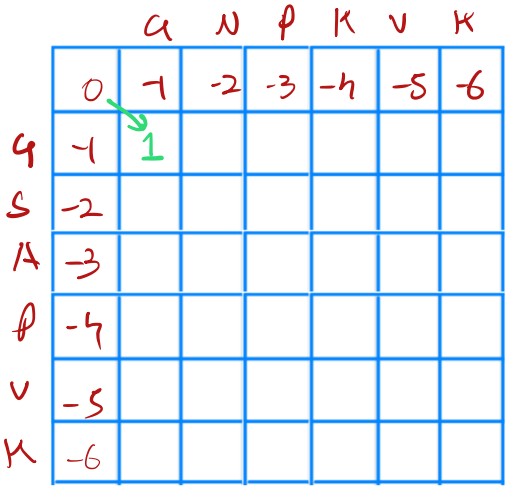
all alignments modeled as paths through board

match 1
mismatch 0
gap -1

diagonal => match/mismatch -> +1 or +0
* design board so you can have max score

Gap moves -> ignore game board
-> pay gap penalty
✓ included for DP Table





zeros :: these numbers will never get picked up

do not need game board for seq align
⇒ just go through DP table
⇒ if match use match score

if given substitution matrix
use that to decide scores (game board)

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{cases}$$

$S_{i,j}$ ⇑ DP table

$s(x_i, y_j)$ ↑ substitution matrix (game board)

$S_{i,j-1} + g$: store in var, find max

gameboard needs gap penalty
⇒ fill in 1st row & col of DP Table
⇒ begin for loop in DP[1,1] not DP[0,0]

Semi-global alignment (free end gaps)

=> DONT penalize gaps in last row & last column
(→) (↓)

shift sequences

or

-> let algorithm run to end

-> identify max val in last row & col => max align score

-> back-track from there

local alignment

no penalties for both seq

endgaps -> outer rows, columns

-> to align shorter sequences, get to start for free, reach end

-> local align => start anywhere, stop anywhere

change

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + s(x_i, y_i) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \\ 0 \end{cases}$$

new starting points

↓

move can start anywhere on board

-> fill board

-> find highest value in entire board

=> That is maximum local align score

=> back-tracking stops at zeros

multiple LA

-> run algorithm

-> set max path = 0 => no longer allowed

=> find next max => select

↑

set a threshold for maxima

Game Board (Scoring Matrix/Dictionary)

Start

~~Row 1~~

~~Col 1~~

Dynamic Programming Table

3	7	8	8	7	6	8
5	3	7	4	2	7	3
6	7	1	8	8	6	2
5	7	7	5	3	5	7
4	1	4	1	10	7	5
6	3	10	2	3	7	5
10	3	1	8	8	7	7

3 options for [1,1]

- Diag
- Down
- Right

$S_{11} = \max \begin{cases} S_{00} + s_{11} \\ S_{01} + g \\ S_{10} + g \end{cases}$ (DP Table, score mat)

$S_1 = \max \begin{cases} 3 + 3 \\ 2 - 1 \\ 2 - 1 \end{cases} \Rightarrow$ Diag

3	2	1	0	-1	-2	-3
2	6					
1						
0						
-1						
-2						
-3						

Row 1 & Col 1 are empty for sequences

Code:

```
for i in range(0, r):
    for j in range(0, c):
```

		G	C	A	G	T
	0	0	0	0	0	0
T	0	-2	-2	-2	-2	2
G	0	2	-2	-2	2	-2
T	0	-2	-2	-2	-2	2
C	0	-2	2	-2	-2	-2
G	0	2	-2	-2	2	-2

		A	C	C	G
	0	-2	-4	-6	-8
A	-2	2	0	-2	-4
C	-4	0	4	2	
A	-6				
G	-8				

match = 2
mismatch = 0
gap = -2

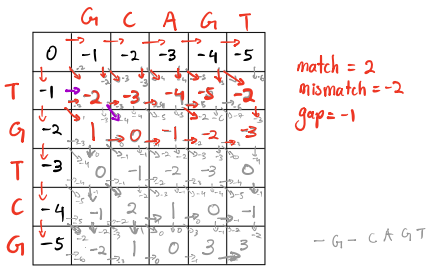
		G	C	A	G	T
	0	0	0	0	0	0
T	0					
G	0					
T	0					
C	0					
G	0					

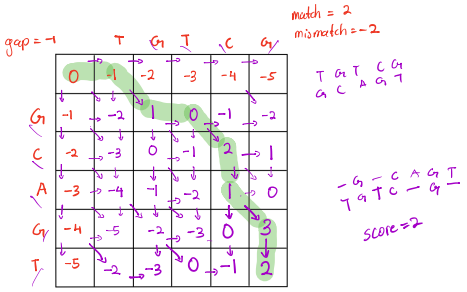
		A	C	C	G
	0	-2	-4	-6	-8
A	-2	2	0	-2	-4
C	-4	0	4	2	0
A	-6	-2	2	4	2
G	-8	-4	0	2	6

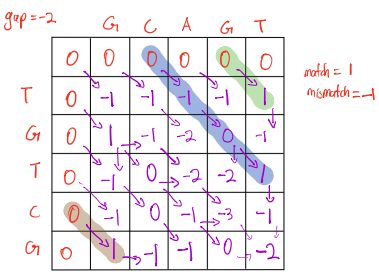
match = 2
mismatch = 0
gap = -2

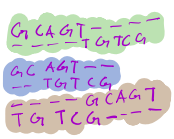
score = 2

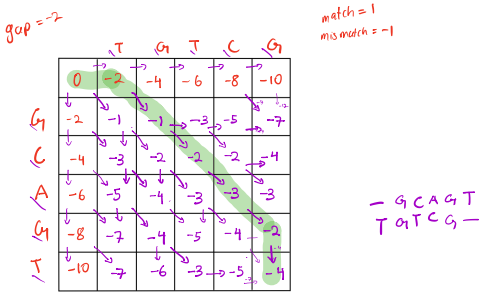
A C C G
A CA G

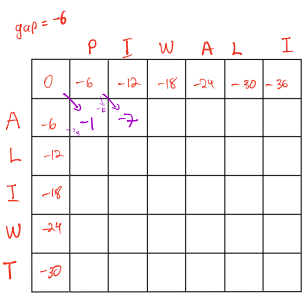


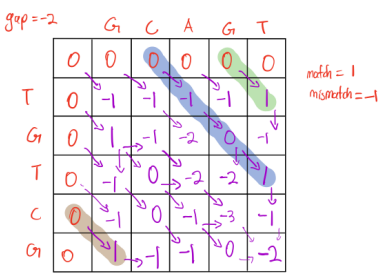


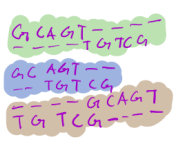












Ask Dr. Sacan

are the three alignments valid?

gap = -2

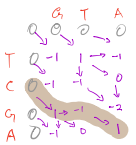
		G1	C	A	G1	T
	0	0	0	0	0	0
T	0	0	0	0	0	1
G1	0	1	0	0	1	0
T	0	0	0	0	0	2
C	0	0	1	0	0	0
G1	0	1	0	0	1	0

match = 1
mismatch = -1

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + s(x_i, y_i) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \\ 0 \end{cases}$$

- T G T C G
G C A G T -

free end gap



match = 1
mismatch = -1
gap = -2

T C G1 - A
- - G1 T A

Sum of all Pairs alignment

a M W G1 F N W
b M R - - - W
c M - G1 F - N

$S_{\text{all pair}} = S_{ab} + S_{ac} + S_{bc}$

= -23

a M W G1 F N W
b M R - - - W
5 -3 -5 -5 -5 11 -2

a M W G1 F N W
c M - G1 F - N
5 -5 6 6 -5 -4 3

b M R - - - W
c M - G1 F - N
5 -5 -5 -5 -5 -5 -4 -24

Entropy

ACA
AGT
TAT
GTT

$f_{1,A} = \frac{2}{4} = 0.5$

$f_{1,T} = \frac{1}{4} = 0.25$

$f_{1,G} = \frac{1}{4} = 0.25$

$f_{1,C} = 0$

number of unrooted tree