Sequence Alignment Algorithms

by Ahmet Sacan

Some images are from "Understanding Bioinformatics" and Wikipedia.

Finding optimal alignment

- Given two sequences, find an alignment that gives maximum alignment score.

```
GS-APVK     -GSAP-VK     GSA-P-VK     ...
G-NPKVK     G-N-PKVK     G--NPKVK     ...
score:3     score:2      score:4      ...
```

- Two sequences with 100 residues can be aligned in ~10^75 different ways.
- Two sequences with 1000 residues can be aligned in ~10^600 different ways.

Dynamic Programming is about re-using solutions to subproblems

- Game: Start from top left corner, move right, down, or diagonal until you reach bottom right. Collect money on diagonal moves. Pay $1 penalty on non-diagonal moves and not collect the money on the target square.

5	2	5
6	3	4
3	12	2

3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

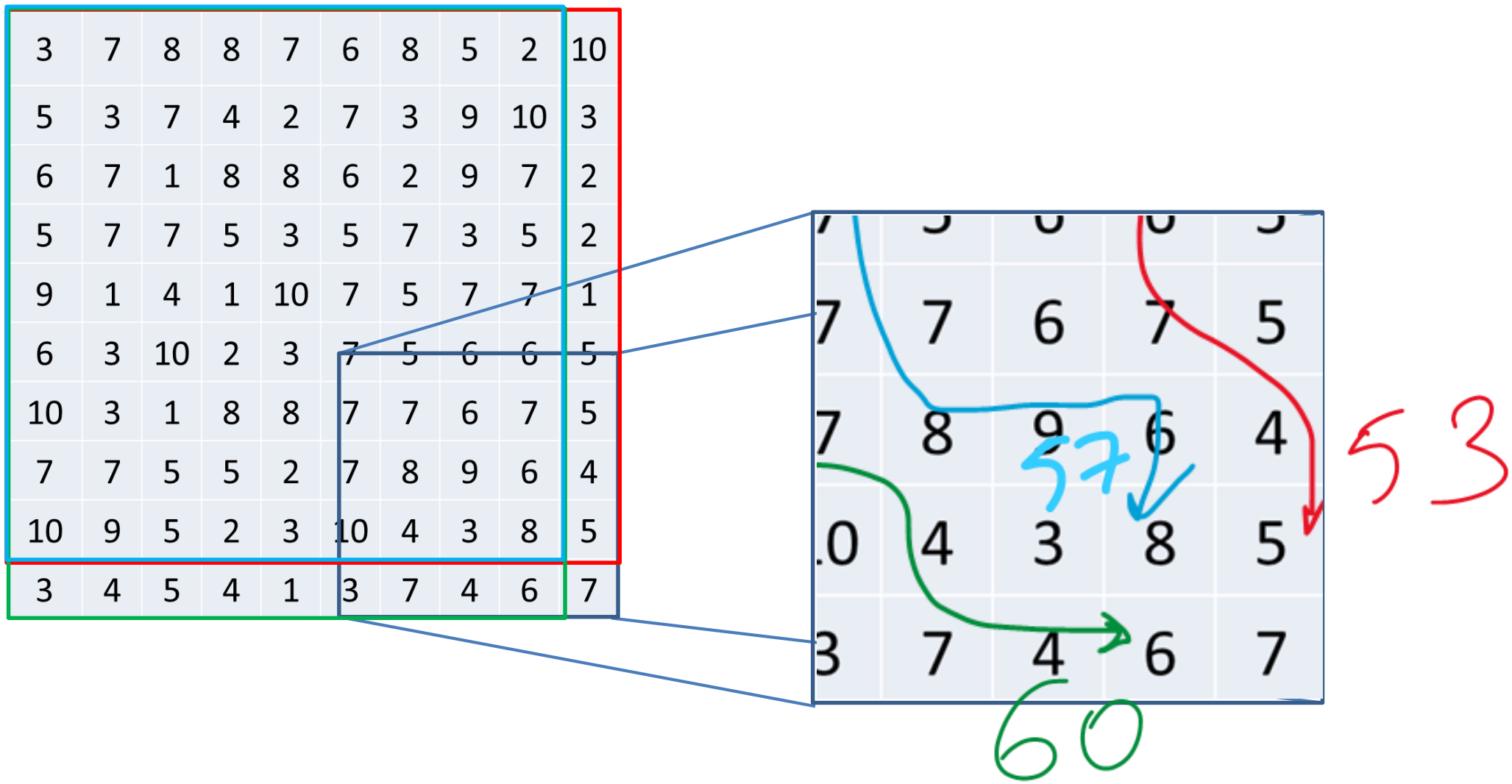
3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

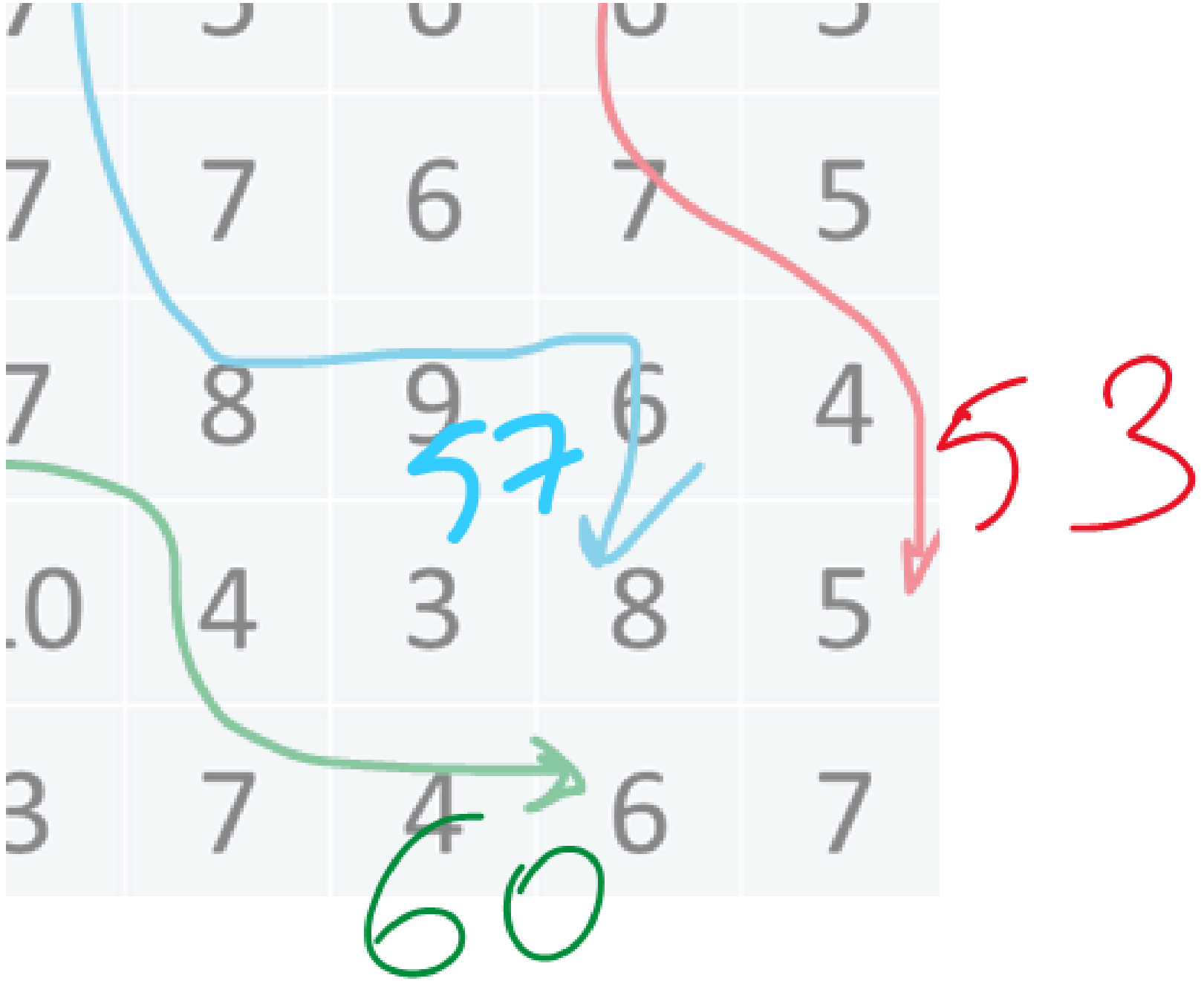
3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

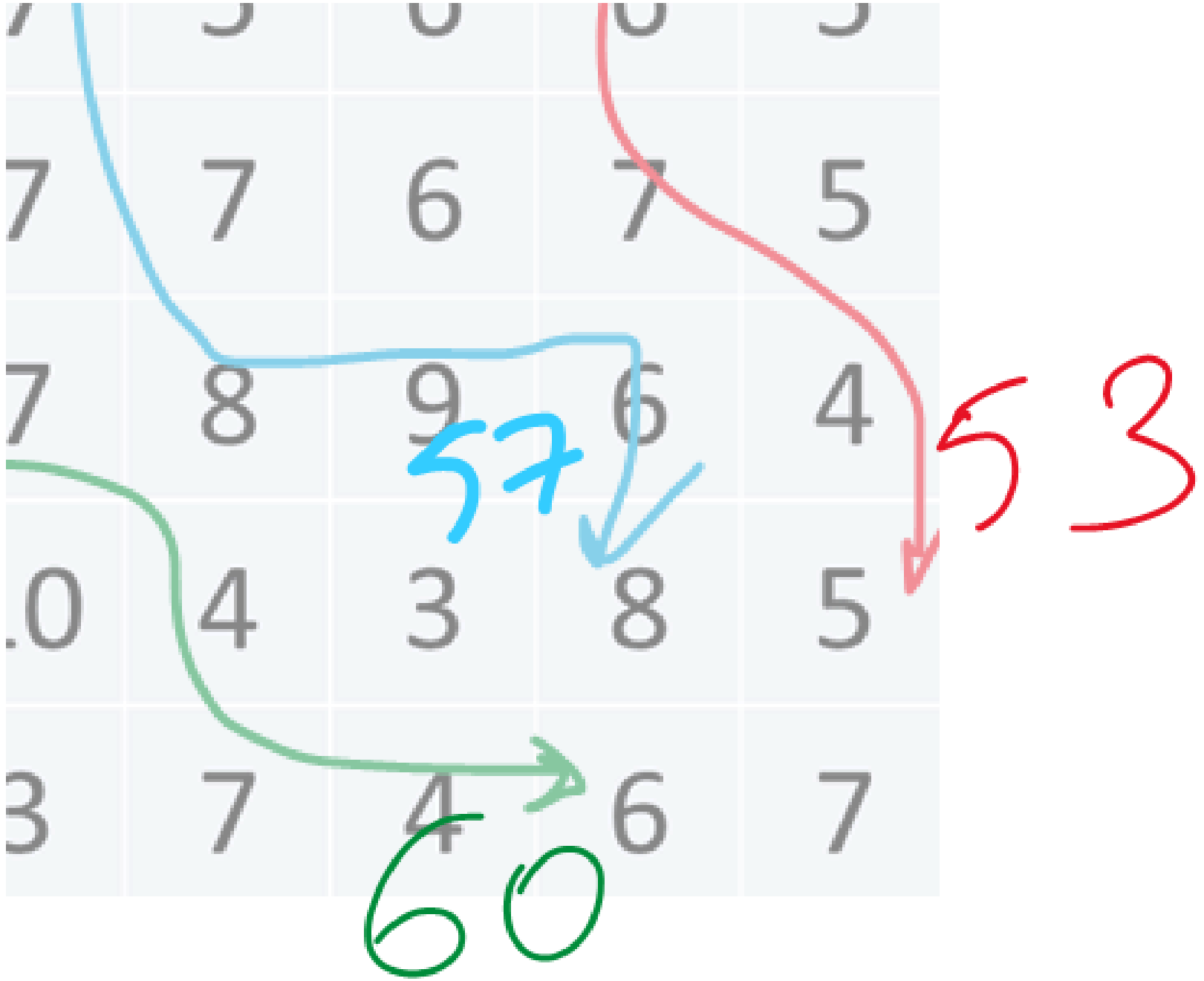
3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

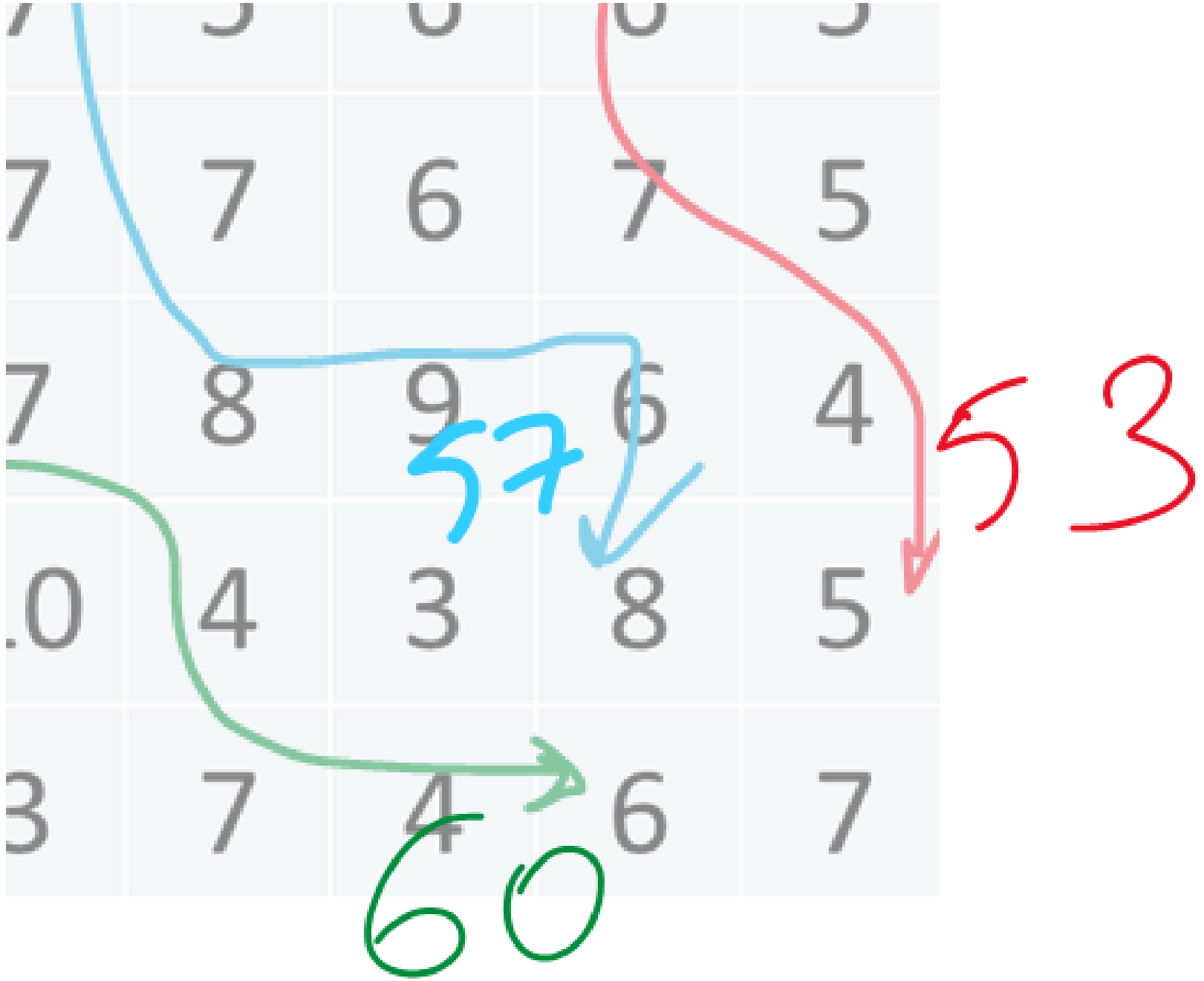
3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

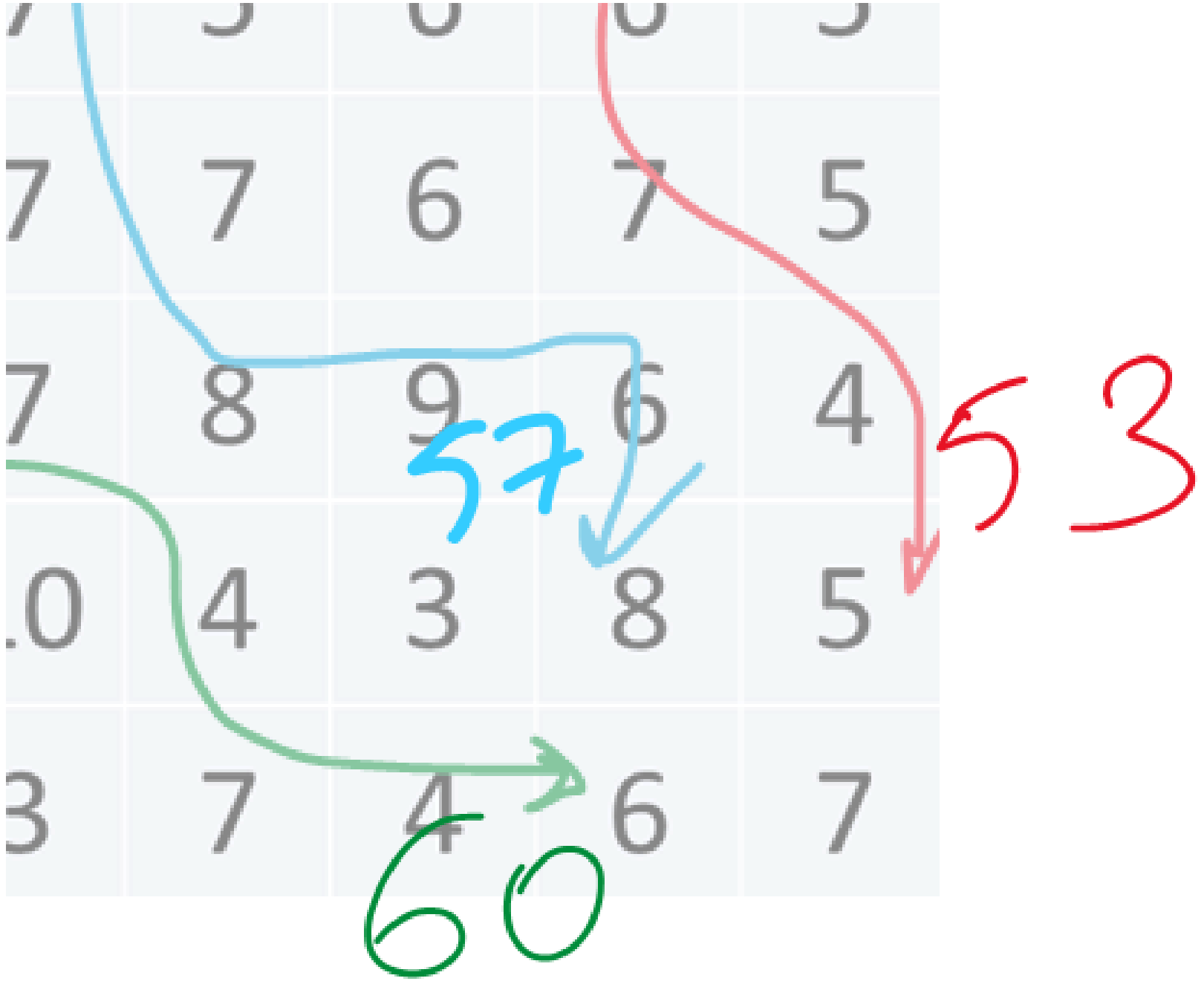
3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7


3 7 8 8 7 6 8 5 2 10
5 3 7 4 2 7 3 9 10 3
6 7 1 8 8 6 2 9 7 2
5 7 7 5 3 5 7 3 5 2
9 1 4 1 10 7 5 7 7 1
6 3 10 2 3 7 5 6 6 5
10 3 1 8 8 7 7 6 7 5
7 7 5 5 2 7 8 9 6 4
10 9 5 2 3 10 4 3 8 5
3 4 5 4 1 3 7 4 6 7
7 7 6 7 5
7 8 9 6 4
.0 4 3 8 5
3 7 4 6 7
57
53
60


7 7 6 7 5
7 8 9 6 4
.0 4 3 8 5
3 7 4 6 7
57
53
60


7 7 6 7 5
7 8 9 6 4
.0 4 3 8 5
3 7 4 6 7
57
53
60


7 7 6 7 5
7 8 9 6 4
.0 4 3 8 5
3 7 4 6 7
57
53
60


7 7 6 7 5
7 8 9 6 4
.0 4 3 8 5
3 7 4 6 7
57
53
60

3	7	8	8	7	6	8	5	2	10
5	3	7	4	2	7	3	9	10	3
6	7	1	8	8	6	2	9	7	2
5	7	7	5	3	5	7	3	5	2
9	1	4	1	10	7	5	7	7	1
6	3	10	2	3	7	5	6	6	5
10	3	1	8	8	7	7	6	7	5
7	7	5	5	2	7	8	9	6	4
10	9	5	2	3	10	4	3	8	5
3	4	5	4	1	3	7	4	6	7

Game Board

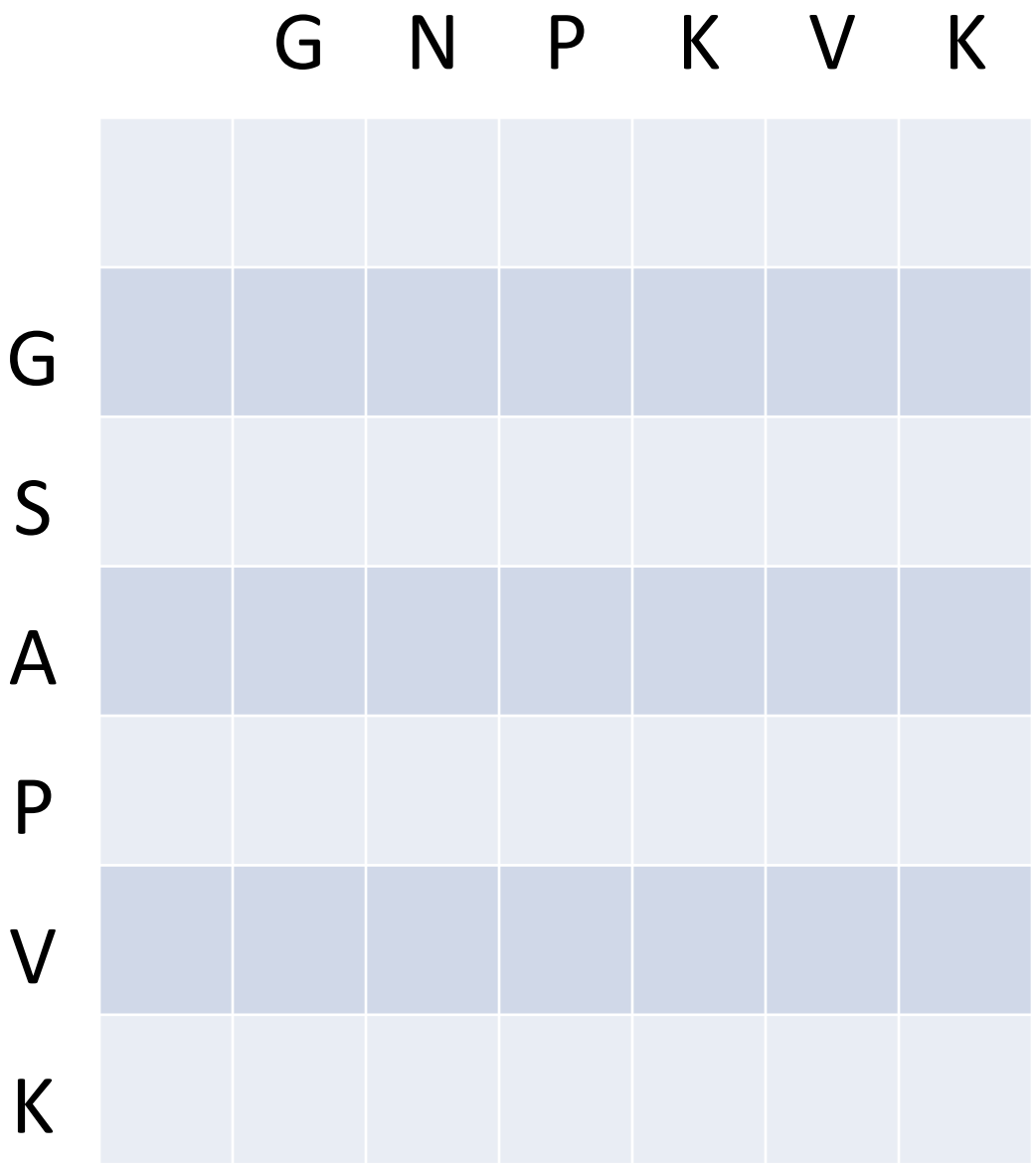
5	2	5
6	3	4
3	12	2

Dynamic Programming Table

Game Board

GSAPVK

GNPKVK



GS-APVK
G-NPKVK

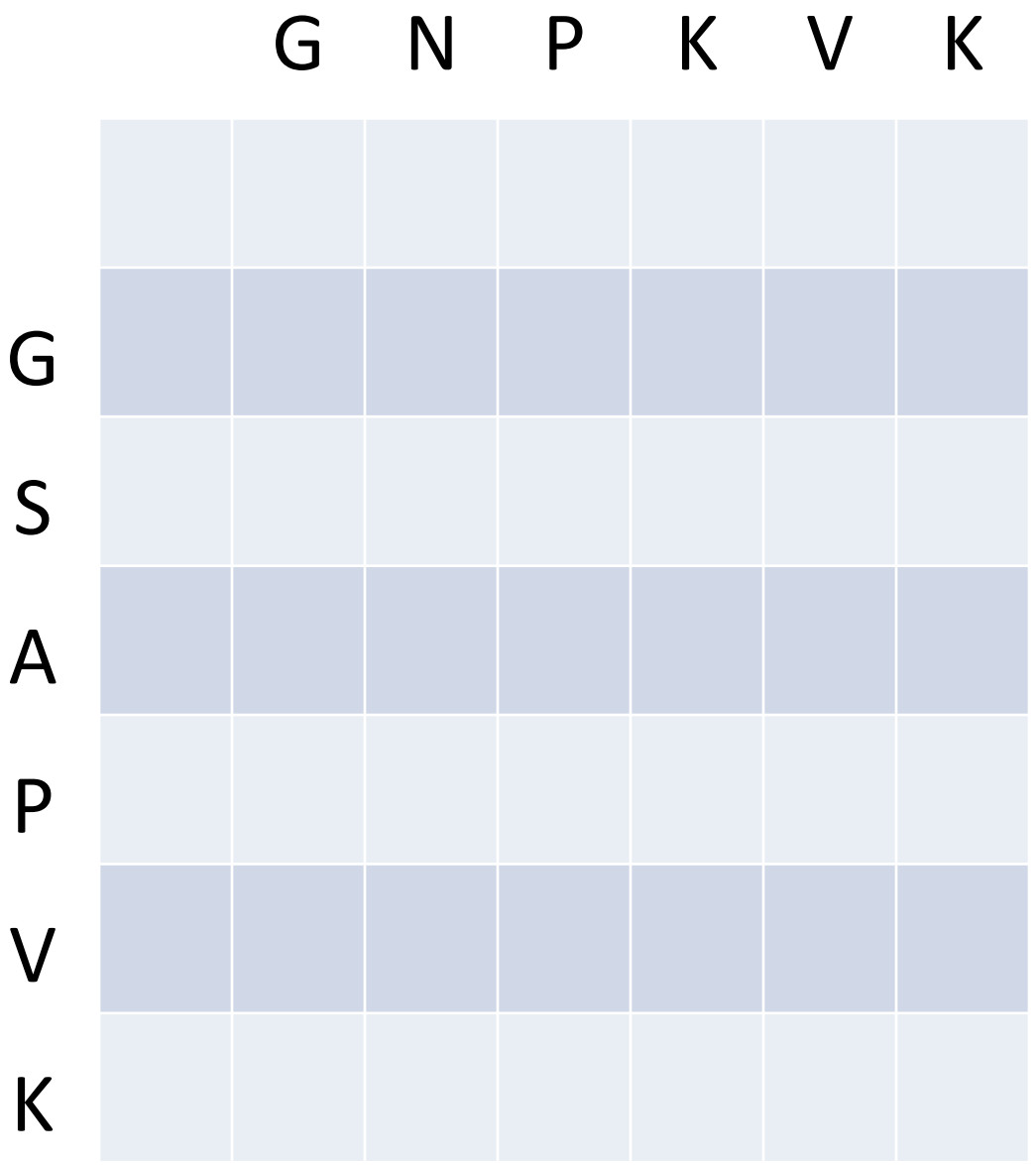
GSAPVK

GNPKVK

match: +1
mismatch: 0

gap: -1

Game Board



GSAPVK

GNPKVK

match: +1
mismatch: 0

gap: -1

Game Board

		G	N	P	K	V	K
	0						
G		1	0	0	0	0	0
S		0	0	0	0	0	0
A		0	0	0	0	0	0
P		0	0	1	0	0	0
V		0	0	0	0	1	0
K		0	0	0	1	0	1

GSAPVK

GNPKVK

match: +1
mismatch: 0

gap: -1

Game Board

		G	N	P	K	V	K
	0	0	0	0	0	0	0
G	0	**1**	0	0	0	0	0
S	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0
P	0	0	0	**1**	0	0	0
V	0	0	0	0	0	**1**	0
K	0	0	0	0	**1**	0	**1**

Game Board

GSAPVK

GNPKVK

match: +1

mismatch:0

gap: -1

		G	N	P	K	V	K
	0	0	0	0	0	0	0
G	0	**1**	0	0	0	0	0
S	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0
P	0	0	0	**1**	0	0	0
V	0	0	0	0	0	**1**	0
K	0	0	0	0	**1**	0	**1**

Dynamic Programming Table

		G	N	P	K	V	K
	0	-1	-2	-3	-4	-5	-6
G	-1	1					
S	-2						
A	-3						
P	-4						
V	-5						
K	-6						

Game Board

GSAPVK

GNPKVK

match: +1

mismatch:0

gap: -1

		G	N	P	K	V	K
	0	0	0	0	0	0	0
G	0	**1**	0	0	0	0	0
S	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0
P	0	0	0	**1**	0	0	0
V	0	0	0	0	0	**1**	0
K	0	0	0	0	**1**	0	**1**

Dynamic Programming Table

		G	N	P	K	V	K
	0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-2	-3	-4
S	-2	0	1	0	-1	-2	-3
A	-3	-1	0	1	0	-1	-2
P	-4	-2	-1	1	1	0	-1
V	-5	-3	-2	0			
K	-6						

Game Board

Dynamic Programming Table

GSAPVK

GNPKVK

match: +1

mismatch:0

gap: -1

		G	N	P	K	V	K
	0	0	0	0	0	0	0
G	0	1	0	0	0	0	0
S	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0
P	0	0	0	1	0	0	0
V	0	0	0	0	0	1	0
K	0	0	0	0	1	0	1

		G	N	P	K	V	K
	0	1	2	3	4	5	-6
G	0	1	0	-1	-2	-3	-4
S	1	1	1	0	-1	-2	-3
A	2	0				-1	-2
P						0	-1
V							
K							

Game Board

GSAPVK

GNPKVK

match: +1

mismatch:0

gap: -1

		G	N	P	K	V	K
	0	0	0	0	0	0	0
G	0	**1**	0	0	0	0	0
S	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0
P	0	0	0	**1**	0	0	0
V	0	0	0	0	0	**1**	0
K	0	0	0	0	**1**	0	**1**

Dynamic Programming Table

		G	N	P	K	V	K
	0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-2	-3	-4
S	-2	0	1	0	-1	-2	-3
A	-3	-1	0	1	0	-1	-2
P	-4	-2	-1	1	1	0	-1
V	-5	-3	-2	0	1		
K	-6						

GSAPVK

GNPKVK

match: +1
mismatch:0

gap: -1

Dynamic Programming Table

		G	N	P	K	V	K
	0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-2	-3	-4
S	-2	0	1	0	-1	-2	-3
A	-3	-1	0	1	0	-1	-2
P	-4	-2	-1	1	1	0	-1
V	-5	-3	-2	0	1	2	1
K	-6	-4	-3	-1	1	1	3

CAG

TACG

match: +3

mismatch:-4

gap: -1

Game Board

		T	A	C	G
	0	0	0	0	0
C	0	-4	-4	**3**	-4
A	0	-4	**3**	-4	-4
G	0	-4	-4	-4	**3**

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1	-2			
A	-2				
G	-3				

CAG

TACG

match: +3

mismatch:-4

gap: -1

Game Board

		T	A	C	G
	0	0	0	0	0
C	0	-4	-4	**3**	-4
A	0	-4	**3**	-4	-4
G	0	-4	-4	-4	**3**

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1	-2			
A	-2				
G	-3				

CAG

TACG

match: +3

mismatch:-4

gap: -1

Game Board

		T	A	C	G
	0	0	0	0	0
C	0	-4	-4	**3**	-4
A	0	-4	**3**	-4	-4
G	0	-4	-4	-4	**3**

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1	-2	-3	1	0
A	-2	-3	1	0	-1
G	-3	-4	0	-1	3

CAG

TACG

Substitution Matrix

	A	C	G	T
A	4	0	1	0
C	0	9	-3	-1
G	1	-3	6	-2
T	0	-1	-2	5

gap: -1

Game Board

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

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1				
A	-2				
G	-3				

CAG

TACG

Substitution Matrix

	A	C	G	T
A	4	0	1	0
C	0	9	-3	-1
G	1	-3	6	-2
T	0	-1	-2	5

gap: -1

Game Board

		T	A	C	G
	0	0	0	0	0
C	0	-1	0	9	-3
A	0	0	4	0	1
G	0	-2	1	-3	6

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1				
A	-2				
G	-3				

CAG

TACG

Substitution Matrix

	A	C	G	T
A	4	0	1	0
C	0	9	-3	-1
G	1	-3	6	-2
T	0	-1	-2	5

gap: -1

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1				
A	-2				
G	-3				

CAG

TACG

Substitution Matrix

	A	C	G	T
A	4	0	1	0
C	0	9	-3	-1
G	1	-3	6	-2
T	0	-1	-2	5

gap: -1

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1	-1	-1	7	6
A	-2	-1	3	6	8
G	-3	-2	2	5	12

CAG

TACG

Substitution Matrix

	A	C	G	T
A	4	0	1	0
C	0	9	-3	-1
G	1	-3	6	-2
T	0	-1	-2	5

gap: -1

Dynamic Programming Table

		T	A	C	G
	0	-1	-2	-3	-4
C	-1	-1	-1	7	6
A	-2	-1	3	6	8
G	-3	-2	2	5	12

Applying Dynamic Programming to sequence alignment

- Score of an aligned amino acid pair is independent of others
- Due to additivity of scores, any sub-alignment of the optimal alignment must also be optimal
- Thus, the optimal score can be found by search for local optima

Optimality of sub-alignments

- If this is the best alignment:

  GS-AQVK

  G-NPKVK

- Then, any part of it, for example

  -AQ

  NPK

must be better than any other possible alignment. Otherwise, say it was not as good as:

  A-Q

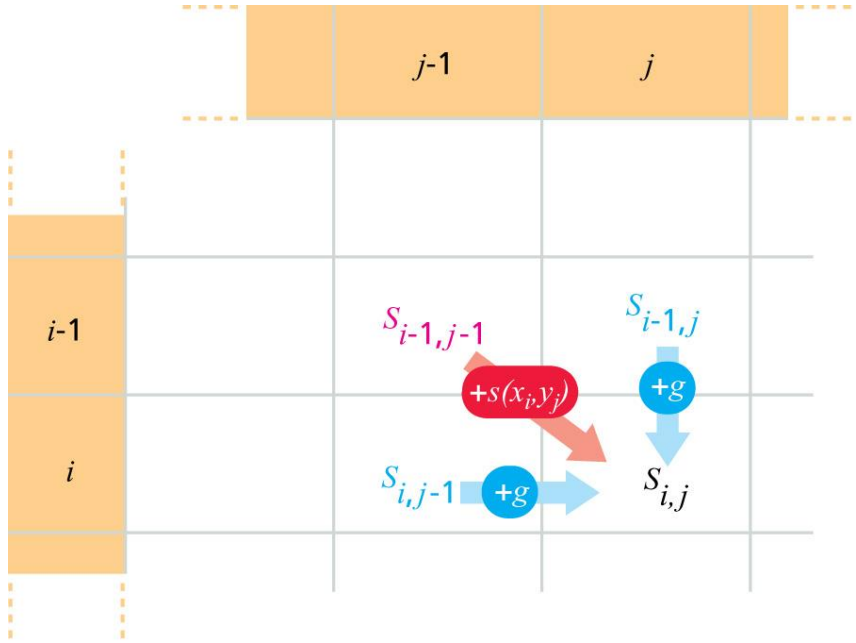
  NPK

  - Becase then, the following would've been the best alignment:

    GSA-QVK

    G-NPKVK

Solving for Sij using subsolutions



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

Exercise

- Find the optimal alignment of:

ACAG

ACCG

- Use gap=-2, and

  match: +2, mismatch:0

Exercise

- Find the optimal alignment of:

ACAG

ACCG

- Use gap=-2, and

  match: +2, mismatch:0

Dynamic Programming Table

		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

Exercise

- Find the optimal alignment of:

ACAG

ACCG

- gap=-2, and the following substitution matrix:

	A	C	G
A	5	-5	0
C	-5	2	5
G	0	5	1

Exercise

- Find the optimal alignment of:

ACAG

ACCG

- gap=-2, and the following substitution matrix:

	A	C	G
A	5	-5	0
C	-5	2	5
G	0	5	1

Dynamic Programming Table

		A	C	C	G
	0	-2	-4	-6	-8
A	-2	5			
C	-4				
A	-6				
G	-8				

When Global alignment is not a good choice

- Consider:
  - GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGT**CCAAGCTATGGTCCCTTAACA**GCAATCGGTCTAACA
  - **CCAACCTATGGTCCCTTAACA**
- Global alignment: score=-40

```
GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGTCCAAGCTATGGTCCCTTAACAGCAATCGGTCTAACA
 | ||    | || |      || |      |    ||||   | |  || | |
-C-CA----C-GC-T------AT-G------G----TCCC---T-T--AA-C-A-------------------------
```

  - +21 matches, -0 mismatch, -61 gaps (-29 end, -32 internal)
- More meaningful alignment: score=-42

```
GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGTCCAAGCTATGGTCCCTTAACAGCAATCGGTCTAACA
                                             ||||:||||||||||||||
----------------------------------------CCAACCTATGGTCCCTTAACA--------------
```

  - 20 matches, -1 mismatch, -61 gaps (all end gaps)

Semi-global alignment (aka "free end gaps")

- Consider:
  - GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGT**CCAAGCTATGGTCCCTTAACA**GCAATCGGTCTAACA
  - **CCAACCTATGGTCCCTTAACA**
- Global alignment: ~~score= 40~~ score=-11

```
GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGTCCAAGCTATGGTCCCTTAACAGCAATCGGTCTAACA
 | ||    | || |      || |      |    ||||   | |  || | |
-C-CA----C-GC-T------AT-G------G----TCCC---T-T--AA-C-A-------------------------
```

  - +21 matches, -0 mismatch, ~~-61 gaps (29 end~~, -32 internal)
- Semi-global alignment: ~~score= 42~~ score=+19

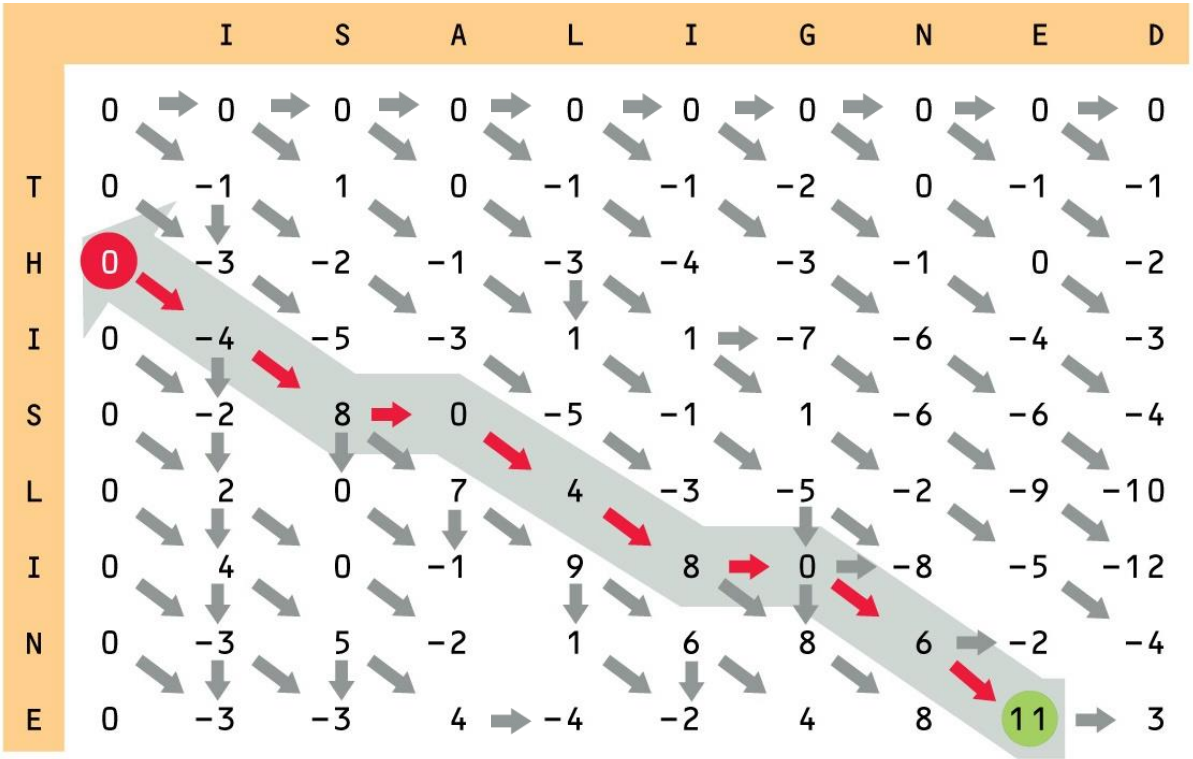
```
GCGCACTTCCGGCATAAAAGGATGGATTTTTGACAATCCCCGATGTCCAAGCTATGGTCCCTTAACAGCAATCGGTCTAACA
                                              ||||:|||||||||||||||
-------------------------------------------CCAACCTATGGTCCCTTAACA-------------
```

  - 20 matches, -1 mismatch, ~~-61 gaps (all end gaps)~~

Semi-global alignment (aka "free end gaps")

1. Use zeros in first row and first column.
2. Use the largest value in the last row or last column as the end point of the alignment.

- BLOSUM62 substitution matrix
- Gap penalty = -8.



When Global alignment is not a good choice - Case 2

- Consider:
  - GCGCACTTCCGGCATAAAA**GGATGGATTTTT**GACAATCCCCGATGTCCAAGCTATGGTCCCTTAACAGCAATCGGTCTAACA
  - CATATCACGTGGTACAAGGTGATTCGTGTCCGCGGGCACCTTGAAGCTTC**GGATGGATTTTT**TGTTGGGACGGCTTTCGTT
- Global alignment:

```
GCGCACT-TC-CG-GCATAAAAGGATGGATTTTTGACAATCCCCGATG-TCC---AAGCT----ATGG-TCCCTTAACAGCAATCGGTCTAACA--
   || | || || |  || |||| | ||||  ||    ||| ||  |  ||   |||||    |||| |   ||    | |  ||| ||  |
---CA-TATCACGTG-GTACAAGG-T-GATTCGTG----TCCGCG-GGCACCTTGAAGCTTCGGATGGATTTTTTGTTGGGA--CGG-CTTTCGTT
```

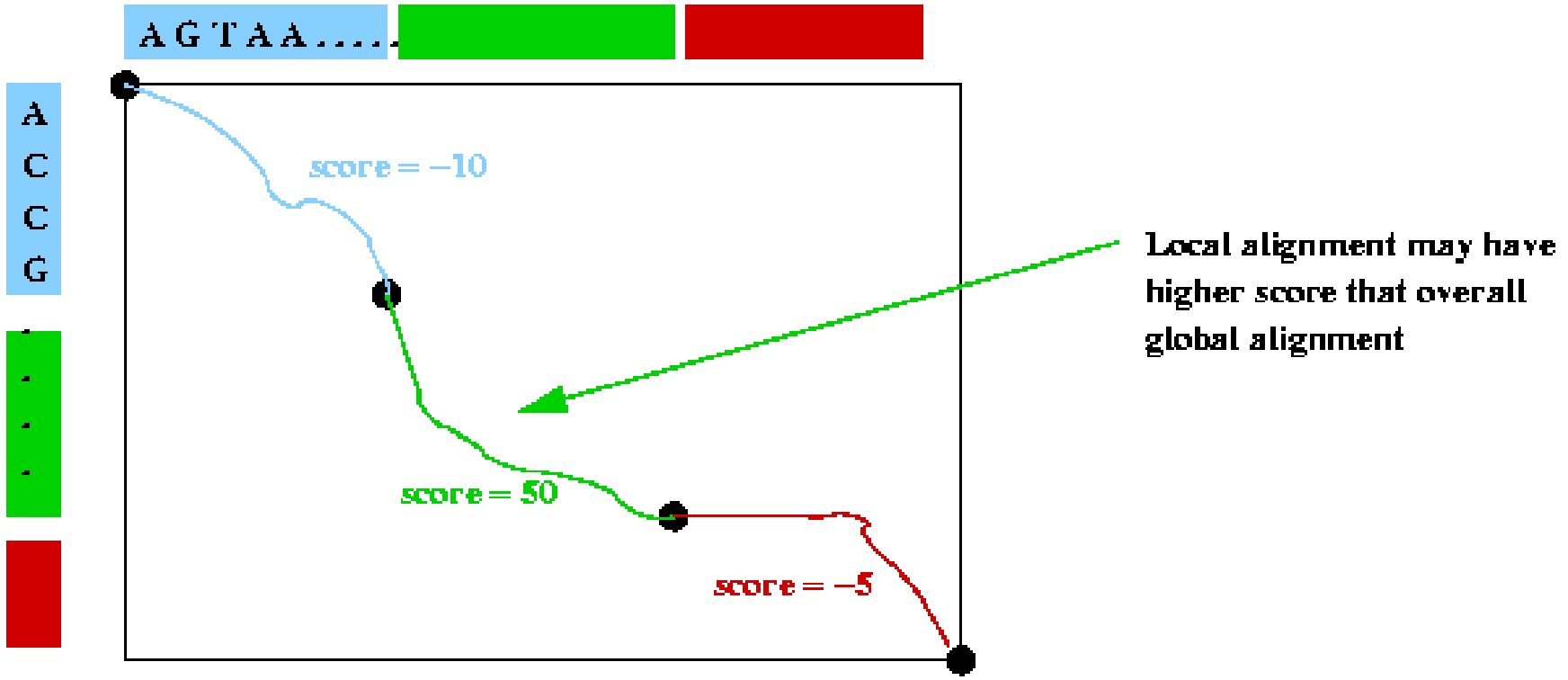
- Semi-Global alignment:

```
GCGCACT-TC-CG-GCATAAAAGGATGGATTTTTGACAATCCCCGATG-TCC---AAGCT----ATGG-TCCCTTA-----ACAGCAATCGGTCTAACA
   || | || || |  || |||| | ||||  ||    ||| ||  |  ||   |||||    |||| |   ||      || ||  ||| |
---CA-TATCACGTG-GTACAAGG-T-GATTCGTG----TCCGCG-GGCACCTTGAAGCTTCGGATGGATTTTTTGTTGGGACGGCTTTCGTT------
```

- More meaningful alignment: Local Alignment

```
GGATGGATTTTT
||||||||||||
GGATGGATTTTT
```

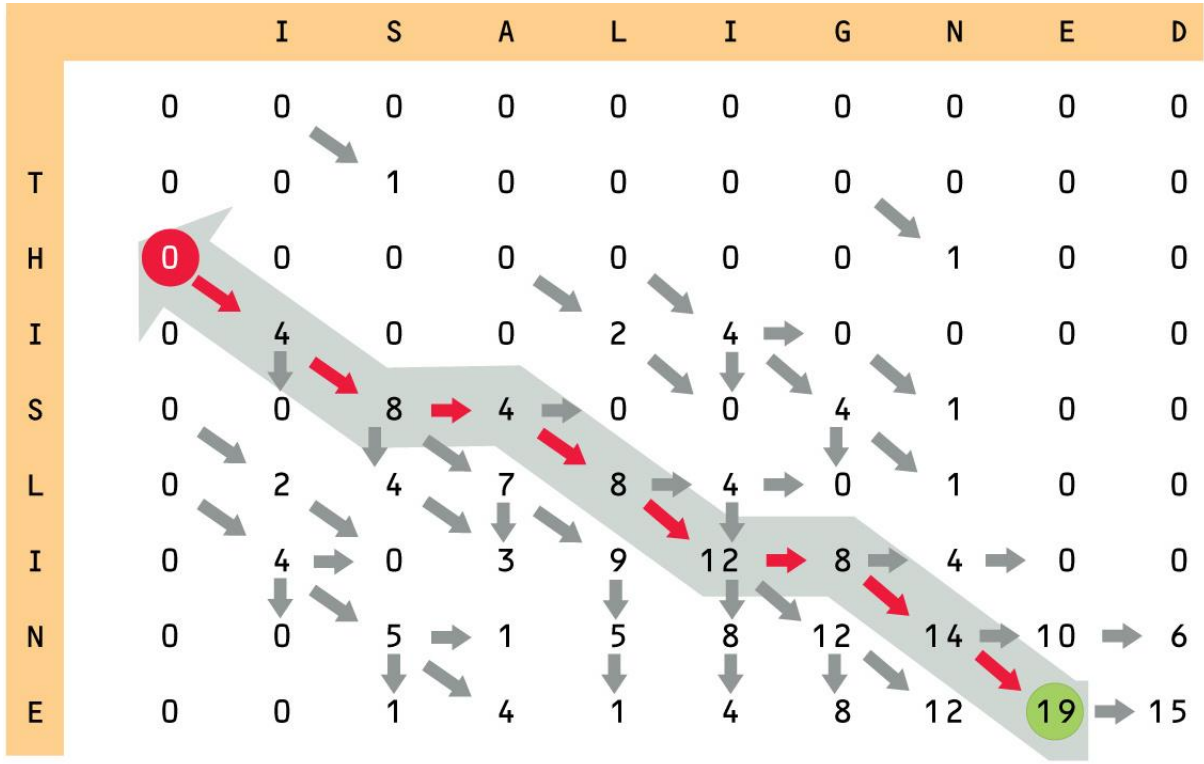
When Global alignment is not a good choice - Case 2



Local alignment

1. Use zeros in first row and first column.
2. $S_{i,j} = max \begin{cases} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \\ \mathbf{0} \end{cases}$
3. Use the largest value anywhere on the table as the end point of the alignment.
4. Stop back-tracking when you reach a 0.

- BLOSUM62 substitution matrix
- Gap penalty = -4.



IS-LI-NE
|| || ||
ISALIGNE

Local Alignment

KVLEFGY

EQLLKALEFKL

match:+4
mismatch:-2

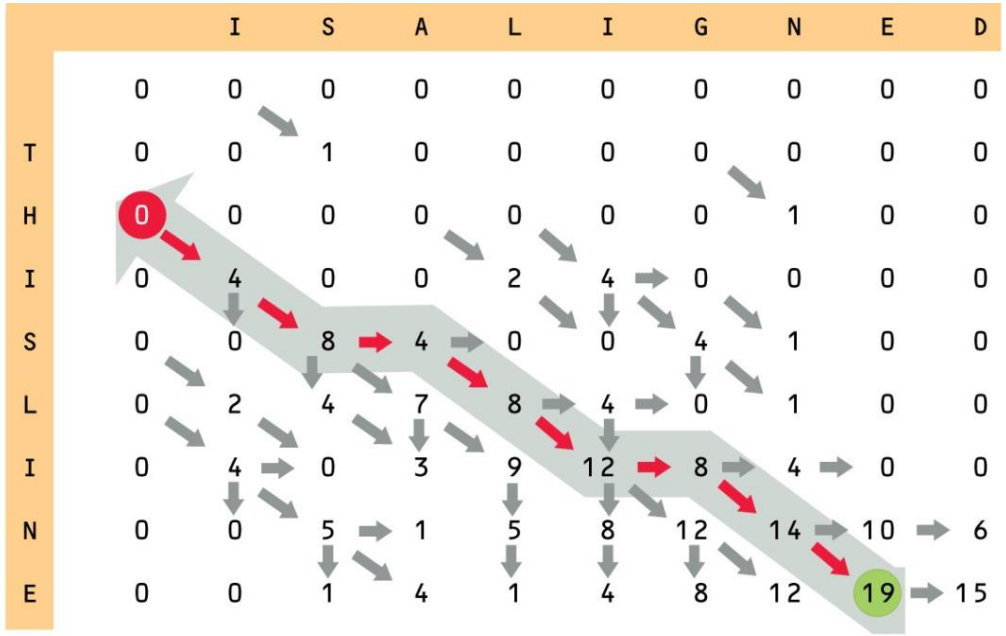
gap:-1

		E	Q	L	L	K	A	L	E	F	K	L
-	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	4	3	2	1	0	4	3
V	0	0	0	0	0	3	2	1	0	0	3	2
L	0	0	0	4	4	3	2	6	5	4	3	7
E	0	4	3	3	3	2	1	5	10	9	8	7
F	0	3	2	2	2	1	0	4	9	14	13	12
G	0	2	1	1	1	0	0	3	8	13	12	11
Y	0	1	0	0	0	0	0	2	7	12	11	10

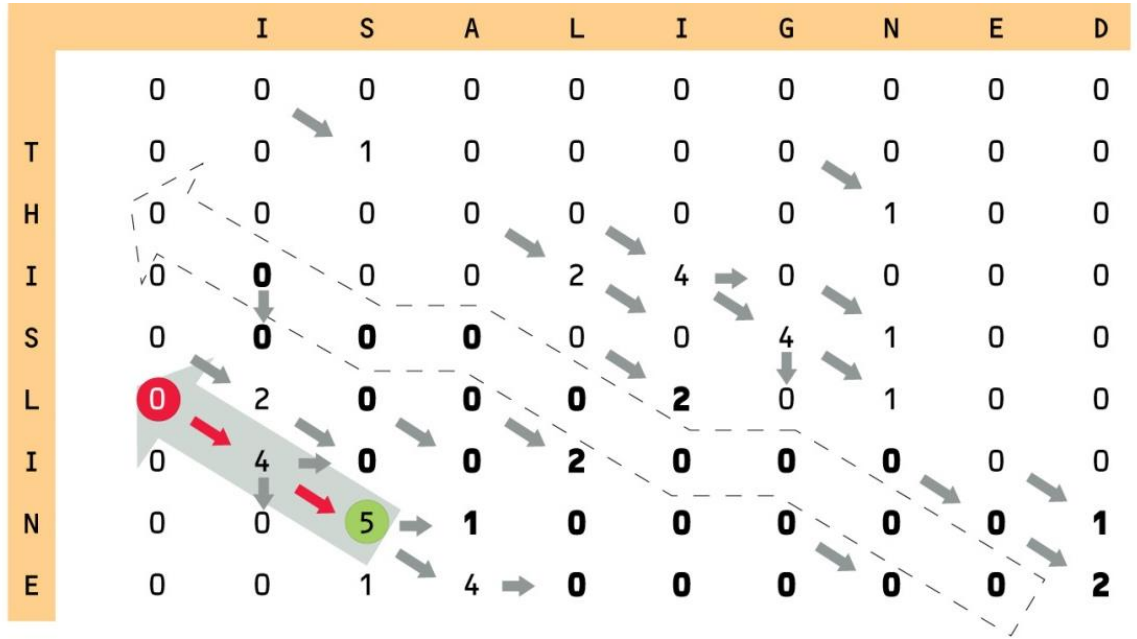
KA-LEF

K-VLEF

Multiple Local alignments



		I	S	A	L	I	G	N	E	D
	0	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	4	0	0	2	4	0	0	0	0
S	0	0	8	4	0	0	4	1	0	0
L	0	2	4	7	8	4	0	1	0	0
I	0	4	0	3	9	12	8	4	0	0
N	0	0	5	1	5	8	12	14	10	6
E	0	0	1	4	1	4	8	12	19	15



		I	S	A	L	I	G	N	E	D
	0	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	0	0	0	2	4	0	0	0	0
S	0	0	0	0	0	0	4	1	0	0
L	0	2	0	0	0	2	0	1	0	0
I	0	4	0	0	2	0	0	0	0	0
N	0	0	5	1	0	0	0	0	0	1
E	0	0	1	4	0	0	0	0	0	2

```
IS-LI-NE
|| || ||
ISALIGNE
```

```
IN
|
IS
```

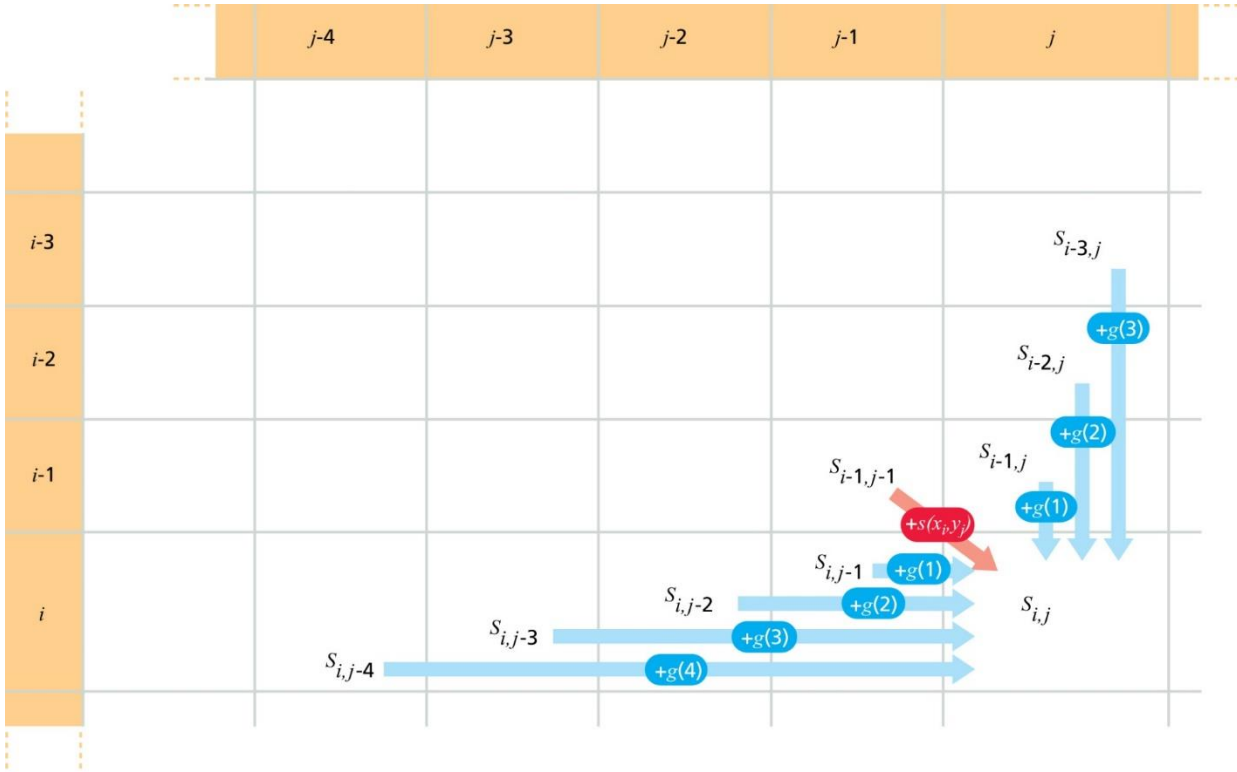
Complex gap function can also be handled efficiently

- Linear gap penalty

$$gappenalty = -n_{gaps} * E$$

- Affine gap penalty

$$gappenalty = -n_{gapopen} * I - n_{gapextend} * E$$



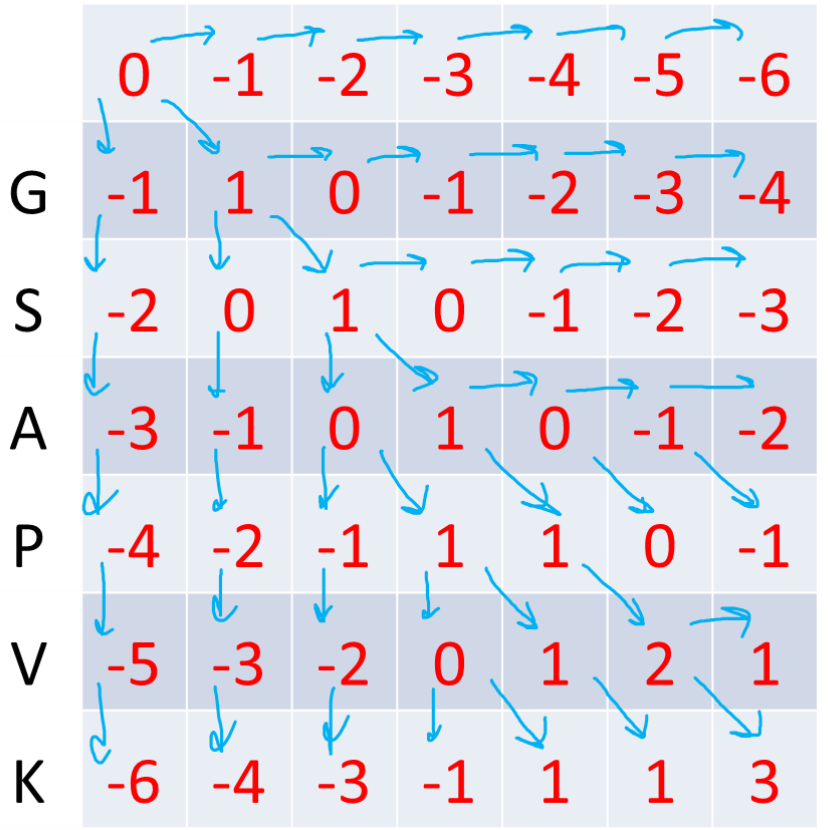
GCGCACTTCC--

G--CA---CGG

Computational Complexity of Alignment

- Lenghts of sequences: m, n
  - O(mn) time
  - O(mn) space

Dynamic Programming Table



Computational Complexity of Alignment

- If only alignment score is needed (and not the alignment itself):
  - O(mn) time
  - O(min(m, n)) space

Summary

- Sequence alignment problem can be solved efficiently using dynamic programming algorithm.
  - Optimal alignments of shorter sequences are used to find the optimal alignment of the longer sequences.
- Semi-global, local, and multiple-local alignments can be calculated by modifying the global alignment algorithm.
- Match/mismatch scores vs. substitution matrix can be used for scoring character pairings
- Linear or affine gap penalty can be used to score gaps in the alignment.