Semi-Global Alignment

What if:

1. Gaps were not penalized at the start of string 1
2. Gaps were not penalized at the start of string 2
3. Gaps were not penalized at the end of string 1
4. Gaps were not penalized at the end of string 2
5. Any combination of the above?

Suppose that there was no charge for end gaps, that is, all 4 conditions above hold. What would the score of the following alignment be?
(match = +1, mismatch = –1, gap = –2)

```
CCAAGT-CAAGTCGG----
----GTTCAAATCGGGCTT
```

How do we reflect this in our dynamic program?

Semi-Global Alignment

		S	T	R	I	N	G	1
	□	□	□	□	□	□	□	□
S	□	□	□	□	□	□	□	□
T	□	□	□	□	□	□	□	□
R	□	□	□	□	□	□	□	□
I	□	□	□	□	□	□	□	□
N	□	□	□	□	□	□	□	□
G	□	□	□	□	□	□	□	□
2	□	□	□	□	□	□	□	□

What would be different about our computation if we did not charge for gaps at the beginning or end of one string or another?

Semi-Global Alignment

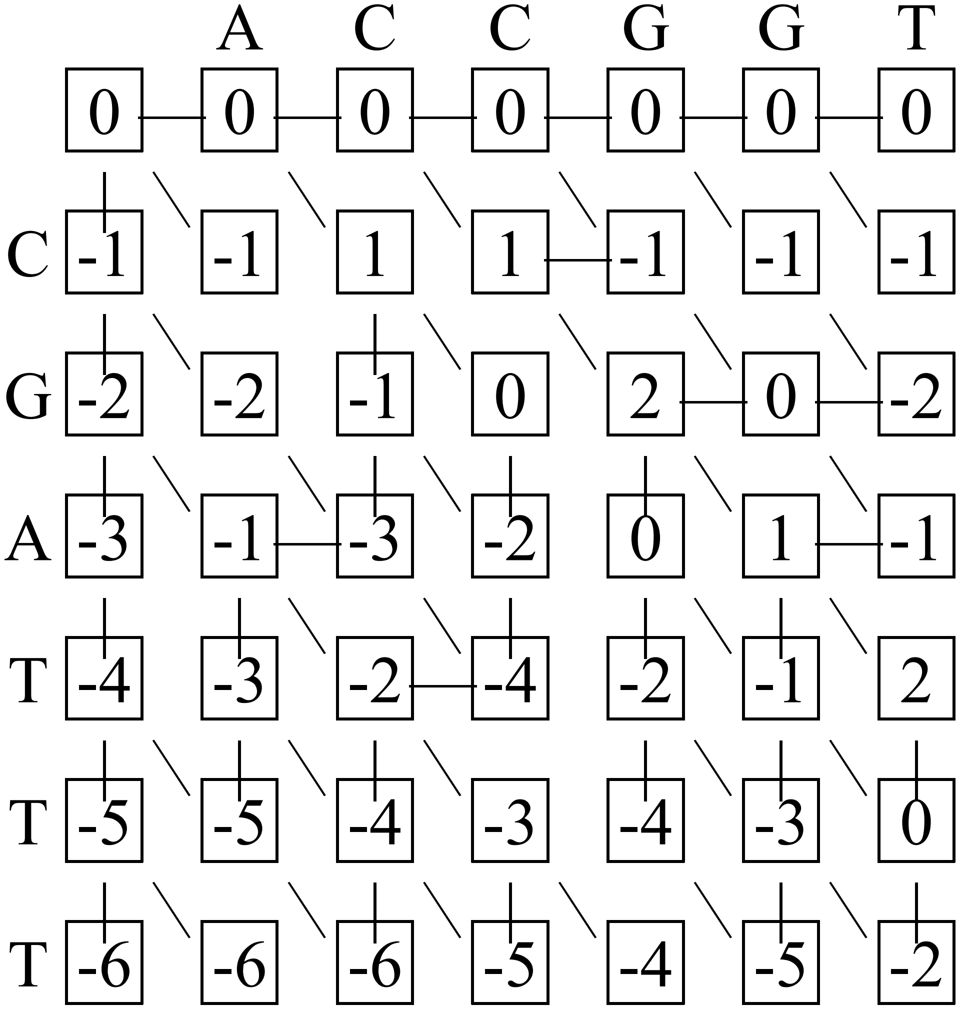
		S	T	R	I	N	G	1
	0	0	0	0	0	0	0	0
S								
T								
R								
I								
N								
G								
2	-7	-8	-3	-4	-5	-4	-5	-6

For free initial gaps in string 2, initialize this row to all “0"s

For free end gaps in string 2, select the greatest element in the last row, and align accordingly

And similarly for dealing with string 1.

Guess my Gap Penalties



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

String1 initial gap penalty: –1
String2 initial gap penalty: 0
Internal gap penalty: –2

Best score with no gap penalty at end of string1:
Best score with no gap penalty at end of string2:

Local Alignment

Given two (unaligned) sequences:

```
ATGCTGACACGTA
ACTACGCTCACAC
```

Select a contiguous substring from each so that their alignment score is as large as possible.

This is called the *local alignment* problem.

Can you find a good local alignment:

match = +1, mismatch = –1, gap = –2

Did you find:

```
GCTGACAC
GCTCACAC
```

Harder or Easier?

Compare the sizes of the search spaces:

Global Alignment:
All possible global alignments of the whole strings

Local Alignment:
All possible global alignments of every pair of substrings, *including* the whole strings

But as you can probably guess, this is a setup for what amounts to a slightly easier, more pleasant algorithm.

Local Alignment Algorithm

It’s easier than global alignment

match = +1, mismatch = –1, gap = –2

Could this be an optimal local alignment of two long sequences:

```
CGTT-AGGGCTTA-C
CAATGAGGGCTTACC
```

No, for two kinds of reason:

- We can lop off stuff at the beginning of the alignment to obtain a better one, because that stuff has negative score.

- Same with the end

Local Alignment Algorithm

Here’s a close-up of that alignment:

C	G	T	T	–	A	G	G	G	C	T	T	A	–	C
C	A	A	T	G	A	G	G	G	C	T	T	A	C	C
+1	–1	–1	+1	–2	+1	+1	+1	+1	+1	+1	+1	+1	–2	+1

Let’s put running totals in the bottom row:

Algorithmically, how could we have discovered that initial section to lop off? Whenever a running total becomes negative, just start over. Set the would-be negative cell to “0.”

C	G	T	T	–	A	G	G	G	C	T	T	A	–	C
C	A	A	T	G	A	G	G	G	C	T	T	A	C	C
+1	–1	–1	+1	–2	+1	+1	+1	+1	+1	+1	+1	+1	–2	+1

Alignment Scoring Tables

Without cutting our losses:

C	G	T	T	–	A	G	G	G	C	T	T	A	–	C
C	A	A	T	G	A	G	G	G	C	T	T	A	C	C
+1	–1	–1	+1	–2	+1	+1	+1	+1	+1	+1	+1	+1	–2	+1
1	0	–1	0	–2	–1	0	1	2	3	4	5	6	4	5

Cutting our losses:

C	G	T	T	–	A	G	G	G	C	T	T	A	–	C
C	A	A	T	G	A	G	G	G	C	T	T	A	C	C
+1	–1	–1	+1	–2	+1	+1	+1	+1	+1	+1	+1	+1	–2	+1
+1	0	0	+1	0	1	2	3	4	5	6	7	8	6	7

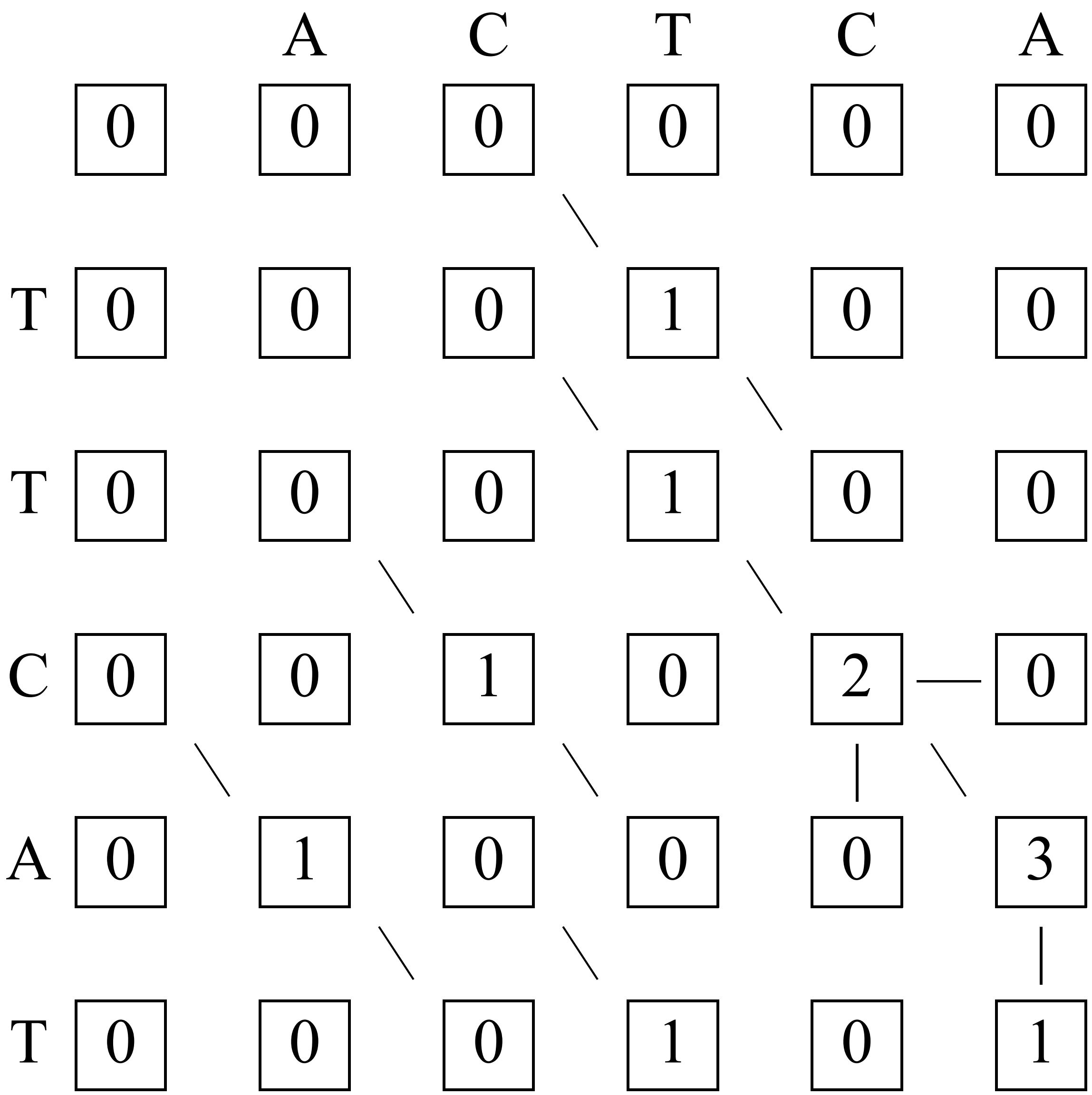
Local Alignment Algorithm

(Smith-Waterman)

		A	C	T	C	A
	□	□	□	□	□	□
T	□	□	□	□	□	□
T	□	□	□	□	□	□
C	□	□	□	□	□	□
A	□	□	□	□	□	□
T	□	□	□	□	□	□

The algorithm is the same as our original alignment algorithm, except that if an entry is about to be negative, we make it “0" instead.

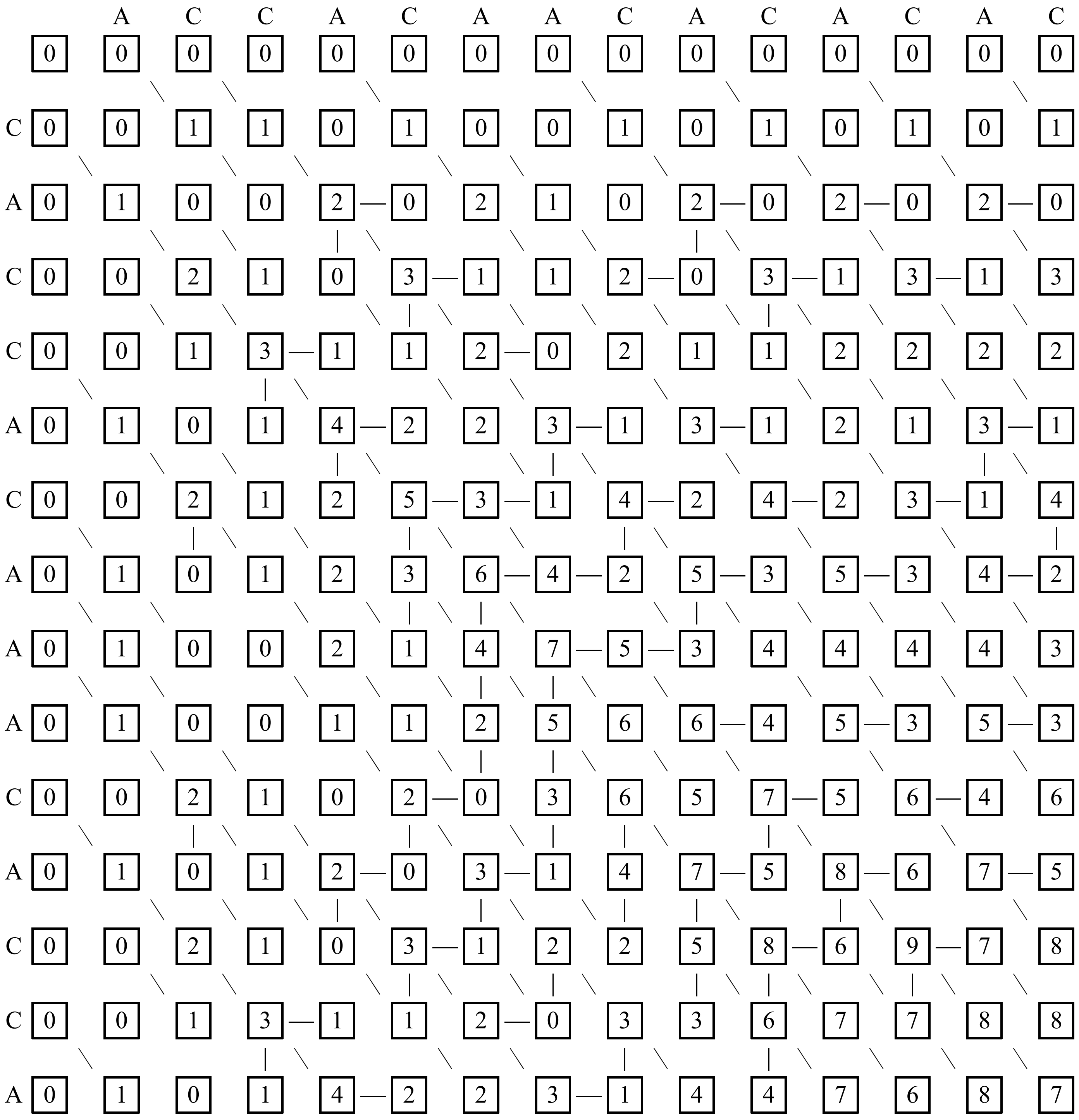
Tracing Back in Smith-Waterman



		A	C	T	C	A
	0	0	0	0	0	0
T	0	0	0	1	0	0
T	0	0	0	1	0	0
C	0	0	1	0	2	0
A	0	1	0	0	0	3
T	0	0	0	1	0	1

To find an optimal alignment, find a largest entry anywhere in the matrix and trace it back, up to but not including a "0."

Tracing Back in Smith-Waterman

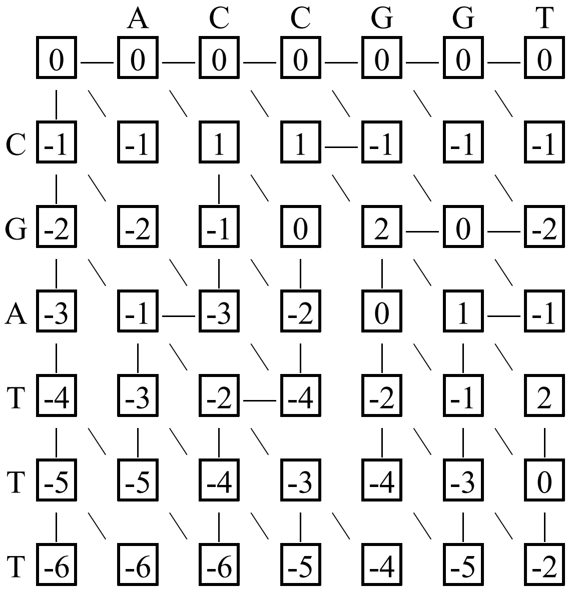


		A	C	C	A	C	A	A	C	A	C	A	C	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	1	0	1	0	0	1	0	1	0	1	0	1
A	0	1	0	0	2	0	2	1	0	2	0	2	0	2	0
C	0	0	2	1	0	3	1	1	2	0	3	1	3	1	3
C	0	0	1	3	1	1	2	0	2	1	1	2	2	2	2
A	0	1	0	1	4	2	2	3	1	3	1	2	1	3	1
C	0	0	2	1	2	5	3	1	4	2	4	2	3	1	4
A	0	1	0	1	2	3	6	4	2	5	3	5	3	4	2
A	0	1	0	0	2	1	4	7	5	3	4	4	4	4	3
A	0	1	0	0	1	1	2	5	6	6	4	5	3	5	3
C	0	0	2	1	0	2	0	3	6	5	7	5	6	4	6
A	0	1	0	1	2	0	3	1	4	7	5	8	6	7	5
C	0	0	2	1	0	3	1	2	2	5	8	6	9	7	8
C	0	0	1	3	1	1	2	0	3	3	6	7	7	8	8
A	0	1	0	1	4	2	2	3	1	4	4	7	6	8	7

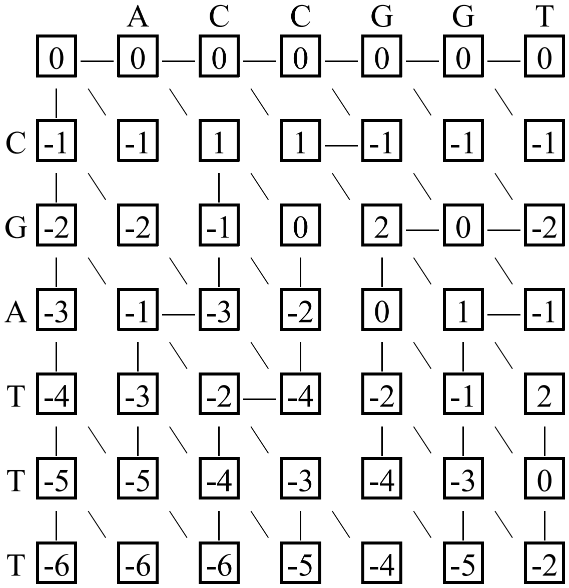
Handout #1 — Varying the Gap Penalties

Here are two copies of the matrix that results when two strings are aligned under a generalized gap penalty function.

1. What is the penalty on a gap at the start of string 1?
2. What is the penalty on a gap at the start of string 2?
3. What is the penalty on an internal gap?
4. What is the optimal score if gaps are not penalized at the end of string 1?
5. What is the optimal score if gaps are not penalized at the end of string 2?
6. What is the optimal score if gaps are not penalized at the end of either string?
7. What is the optimal score if gaps are penalized at the end of the strings?



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



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

Handout #2 — Introduction to Local Alignment

Given two strings, a local alignment is an alignment of two substrings, one taken from each string.

For example, given the two (unaligned) strings

```
ATGCTGACACGTA
ACTACGCTCACAC
```

the following would be a local alignment of score 0:

```
TGCTG
TAC-G
```

What is the best local alignment you can find in those two strings?

Could this be an optimal local alignment of two long sequences:

```
CGTT-AGGGCTTA-C
CAATGAGGGCTTACC
```

Handout #3 — Local Alignment Algorithm

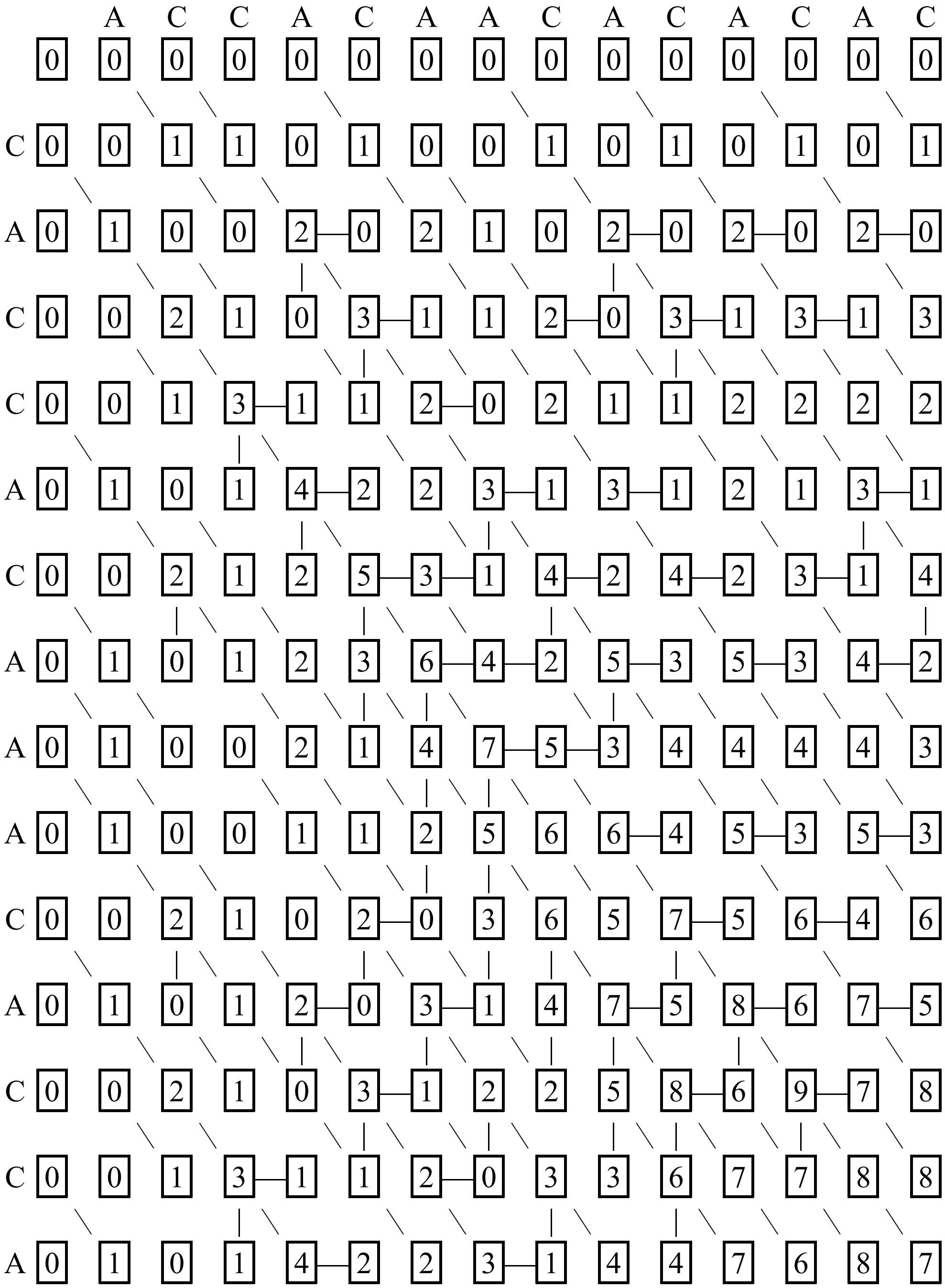
The algorithm is the same as our original alignment algorithm, except that if an entry is about to be negative, we make it “0" instead.

We then obtain our good alignments by finding large entries in the resulting matrix, and tracing them back, up to but not including, a “0” entry.

		A	C	T	C	A
	□	□	□	□	□	□
T	□	□	□	□	□	□
T	□	□	□	□	□	□
C	□	□	□	□	□	□
A	□	□	□	□	□	□
T	□	□	□	□	□	□

Handout #4 — Tracing Back in the Smith Waterman Matrix

Find all optimal local alignments indicated by this scoring matrix.



		A	C	C	A	C	A	A	C	A	C	A	C	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	1	0	1	0	0	1	0	1	0	1	0	1
A	0	1	0	0	2	0	2	1	0	2	0	2	0	2	0
C	0	0	2	1	0	3	1	1	2	0	3	1	3	1	3
C	0	0	1	3	1	1	2	0	2	1	1	2	2	2	2
A	0	1	0	1	4	2	2	3	1	3	1	2	1	3	1
C	0	0	2	1	2	5	3	1	4	2	4	2	3	1	4
A	0	1	0	1	2	3	6	4	2	5	3	5	3	4	2
A	0	1	0	0	2	1	4	7	5	3	4	4	4	4	3
A	0	1	0	0	1	1	2	5	6	6	4	5	3	5	3
C	0	0	2	1	0	2	0	3	6	5	7	5	6	4	6
A	0	1	0	1	2	0	3	1	4	7	5	8	6	7	5
C	0	0	2	1	0	3	1	2	2	5	8	6	9	7	8
C	0	0	1	3	1	1	2	0	3	3	6	7	7	8	8
A	0	1	0	1	4	2	2	3	1	4	4	7	6	8	7

Handout #5 — Smith-Waterman with an Amino Acid Substitution Matrix

Here is the BLOSUM62 scoring matrix:

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	**4**	0	-2	-1	-2	0	-2	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-2	-3	-2
C	0	**9**	-3	-4	-2	-3	-3	-1	-3	-1	-1	-3	-3	-3	-3	-1	-1	-1	-2	-2
D	-2	-3	**6**	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	1	-3	-4	-3
E	-1	-4	2	**5**	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	0	-3	-3	-2
F	-2	-2	-3	-3	**6**	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
G	0	-3	-1	-2	-3	**6**	-2	-4	-2	-4	-3	-2	-2	-2	-2	0	1	0	-2	-3
H	-2	-3	1	0	-1	-2	**8**	-3	-1	-3	-2	1	-2	0	0	-1	0	-2	-2	2
I	-1	-1	-3	-3	0	-4	-3	**4**	-3	2	1	-3	-3	-3	-3	-2	-2	1	-3	-1
J	-1	-3	-1	1	-3	-2	-1	-3	**5**	-2	-1	0	-1	1	2	0	0	-3	-3	-2
L	-1	-1	-4	-3	0	-4	-3	2	-2	**4**	2	-3	-3	-2	-2	-2	-2	3	-2	-1
M	-1	-1	-3	-2	0	-3	-2	1	-1	2	**5**	-2	-2	0	-1	-1	-1	-2	-1	-1
N	-2	-3	1	0	-3	0	-1	-3	0	-3	-2	**6**	-2	0	0	1	0	-3	-4	-2
P	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-1	**7**	-1	-2	-1	1	-2	-4	-3
Q	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	**5**	1	0	0	-2	-2	-1
R	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	**5**	-1	-1	-3	-3	-2
S	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	**4**	1	-2	-3	-2
T	-1	-1	1	0	-2	1	0	-2	0	-2	-1	0	1	0	-1	1	**4**	-2	-3	-2
V	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	-2	**4**	-3	-1
W	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-3	-3	**11**	2
Y	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	**7**

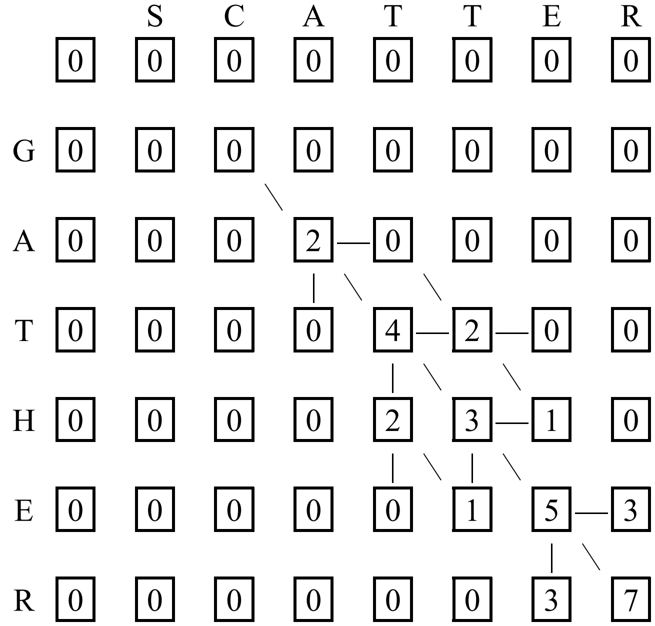
Find an optimal local alignment using the Smith-Waterman algorithm with the scores given above.

		K	A	T	H	Y
	□	□	□	□	□	□
C	□	□	□	□	□	□
H	□	□	□	□	□	□
C	□	□	□	□	□	□
K	□	□	□	□	□	□

Exercises — Local Alignment

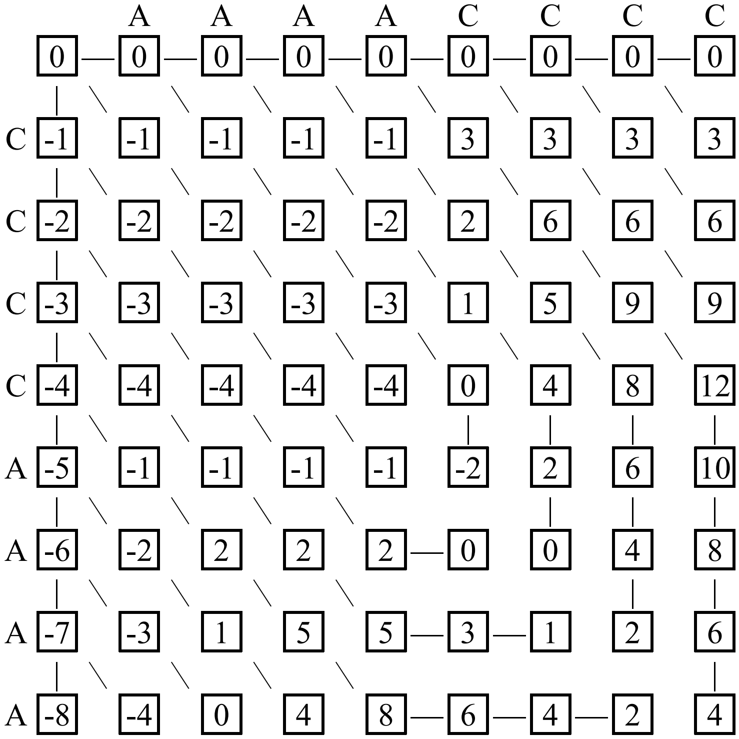
Quick Concepts:

1. What optimal local alignment is suggested by the scoring matrix below?



		S	C	A	T	T	E	R
	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0
A	0	0	0	2	0	0	0	0
T	0	0	0	0	4	2	0	0
H	0	0	0	0	2	3	1	0
E	0	0	0	0	0	1	5	3
R	0	0	0	0	0	0	3	7

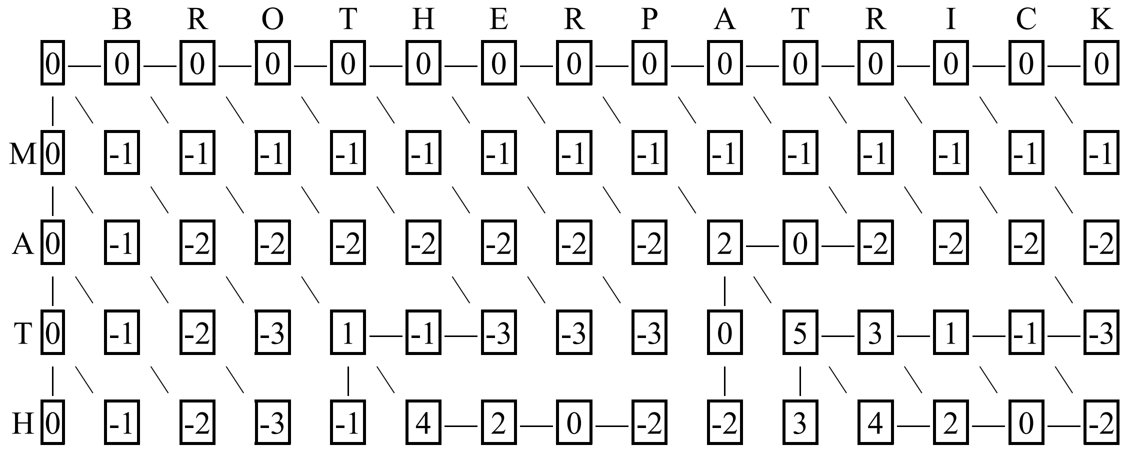
2. What are the various gap, match and mismatch scores in the following global alignment matrix? What are the four optimal alignments, depending on whether end gaps do or don't count for each string?



		A	A	A	A	C	C	C	C
	0	0	0	0	0	0	0	0	0
C	-1	-1	-1	-1	-1	3	3	3	3
C	-2	-2	-2	-2	-2	2	6	6	6
C	-3	-3	-3	-3	-3	1	5	9	9
C	-4	-4	-4	-4	-4	0	4	8	12
A	-5	-1	-1	-1	-1	-2	2	6	10
A	-6	-2	2	2	2	0	0	4	8
A	-7	-3	1	5	5	3	1	2	6
A	-8	-4	0	4	8	6	4	2	4

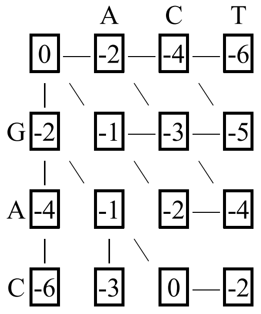
Presentation Problems:

3. This is a scoring matrix for a pair of strings from yesterday:
   a. Does it charge for initial gaps?
   b. How many points are given for a matched column?
   c. How many points are given for a mismatched column?
   d. Give all optimal alignments if there is no charge for initial gaps but there is a charge (of −2) for end gaps
   e. Give all optimal alignments if there is no charge for gaps at the start nor at the end.

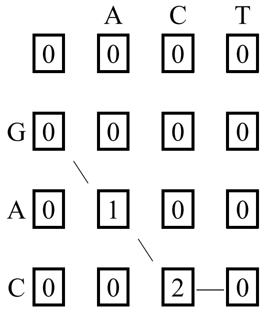


		B	R	O	T	H	E	R	P	A	T	R	I	C	K
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	-1	-2	-2	-2	-2	-2	-2	-2	2	0	-2	-2	-2	-2
T	0	-1	-2	-3	1	-1	-3	-3	-3	0	5	3	1	-1	-3
H	0	-1	-2	-3	-1	4	2	0	-2	-2	3	4	2	0	-2

4. Suppose two sequences are aligned twice, once globally and once locally, under the same match, mismatch and gap scoring. One such example is shown here. Is it the case that the set of edges in the Smith-Waterman matrix (on the right) must be a subset of the edges in the Needleman-Wunsch matrix (on the left)?



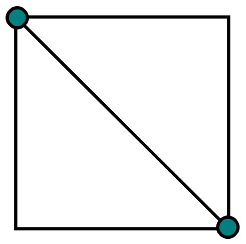
		A	C	T
	0	-2	-4	-6
G	-2	-1	-3	-5
A	-4	-1	-2	-4
C	-6	-3	0	-2



		A	C	T
	0	0	0	0
G	0	0	0	0
A	0	1	0	0
C	0	0	2	0

5. There are three possible global alignments between the strings "A" and "T," shown to the right. These can be enumerated by counting paths from dot to dot on the graph shown below the alignments.

```
A   -A   A-
T   T-   -T
```



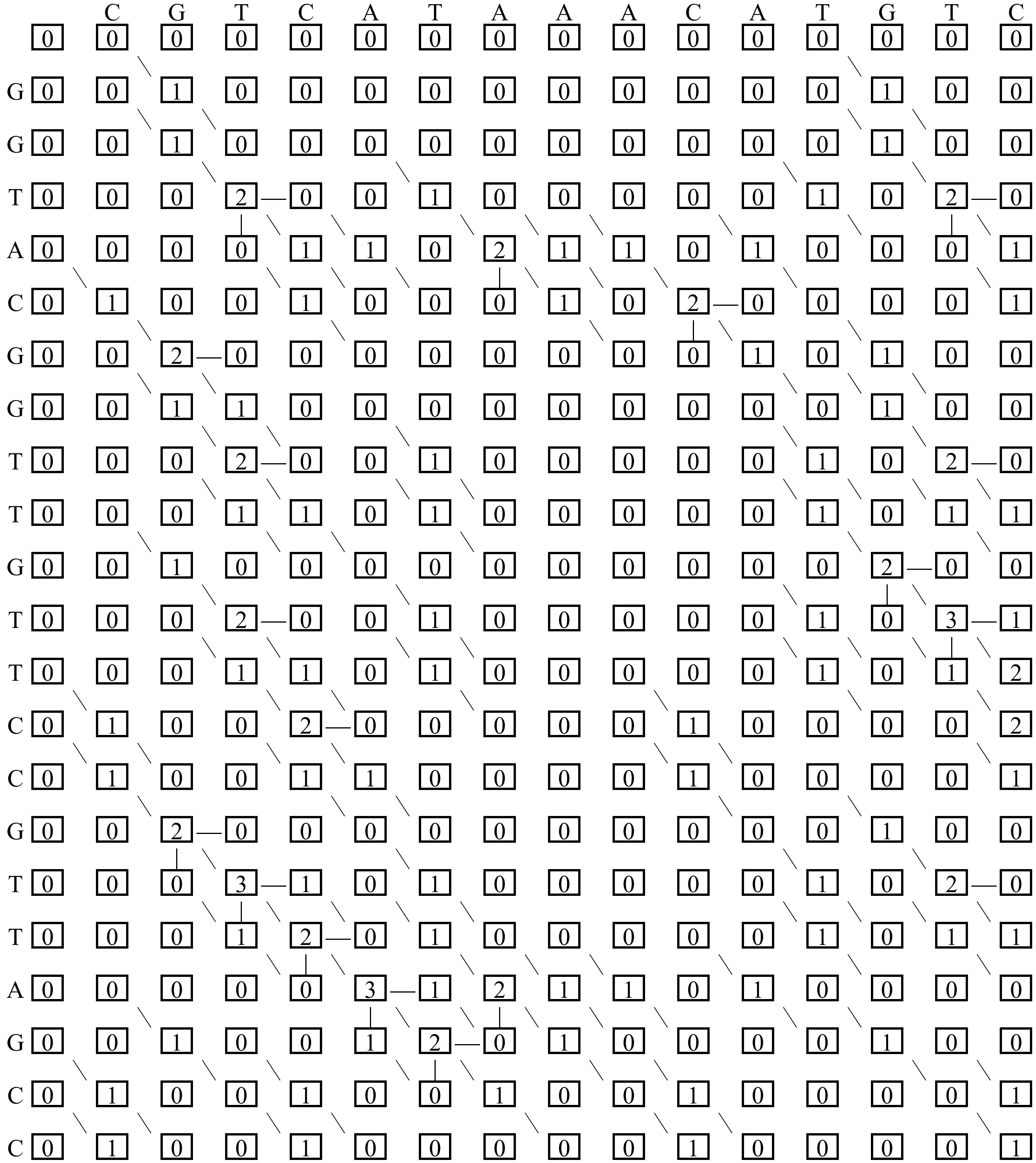
   a. How many global alignments are there between the strings "AC" and "TG"? On what graph would you count paths to answer this question?
   b. Show that there are 129 global alignments between a string of length 3 and a string of length 4.
   c. There are 11 *local* alignments between a string of length 1 and a string of length 2. For example, if the strings were "AT" and "C," then we will have 3 ways to globally align the "A" with the "C," 3 ways to globally align the "T" with the "C," and 5 ways to globally align "AT" with "C." How many local alignments are possible between a string of length 2 and a string of length 3?

6. Find an optimal local alignment using the scoring matrix below, with gap penalty -4.

		W	R	A	P	S
	0	0	0	0	0	0
S	0					
W	0					
I	0					
P	0					
E	0					

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	**4**	0	-2	-1	-2	0	-2	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-2	-3	-2
C	0	**9**	-3	-4	-2	-3	-3	-1	-3	-1	-1	-3	-3	-3	-3	-1	-1	-1	-2	-2
D	-2	-3	**6**	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	1	-3	-4	-3
E	-1	-4	2	**5**	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	0	-3	-3	-2
F	-2	-2	-3	-3	**6**	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
G	0	-3	-1	-2	-3	**6**	-2	-4	-2	-4	-3	-2	-2	-2	-2	0	1	0	-2	-3
H	-2	-3	1	0	-1	-2	**8**	-3	-1	-3	-2	1	-2	0	0	-1	0	-2	-2	2
I	-1	-1	-3	-3	0	-4	-3	**4**	-3	2	1	-3	-3	-3	-3	-2	-2	1	-3	-1
J	-1	-3	-1	1	-3	-2	-1	-3	**5**	-2	-1	0	-1	1	2	0	0	-3	-3	-2
L	-1	-1	-4	-3	0	-4	-3	2	-2	**4**	2	-3	-3	-2	-2	-2	-2	3	-2	-1
M	-1	-1	-3	-2	0	-3	-2	1	-1	2	**5**	-2	-2	0	-1	-1	-1	-2	-1	-1
N	-2	-3	1	0	-3	0	-1	-3	0	-3	-2	**6**	-2	0	0	1	0	-3	-4	-2
P	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-1	**7**	-1	-2	-1	1	-2	-4	-3
Q	-1	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	**5**	1	0	0	-2	-2	-1
R	-1	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	**5**	-1	-1	-3	-3	-2
S	1	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	**4**	1	-2	-3	-2
T	-1	-1	1	0	-2	1	0	-2	0	-2	-1	0	1	0	-1	1	**4**	-2	-3	-2
V	0	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	-2	**4**	-3	-1
W	-3	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-3	-3	**11**	2
Y	-2	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	**7**

7. Find all optimal local alignments for the Local Alignment matrix shown below. What is the optimal score?



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

8. Here are some triangles full of numbers.

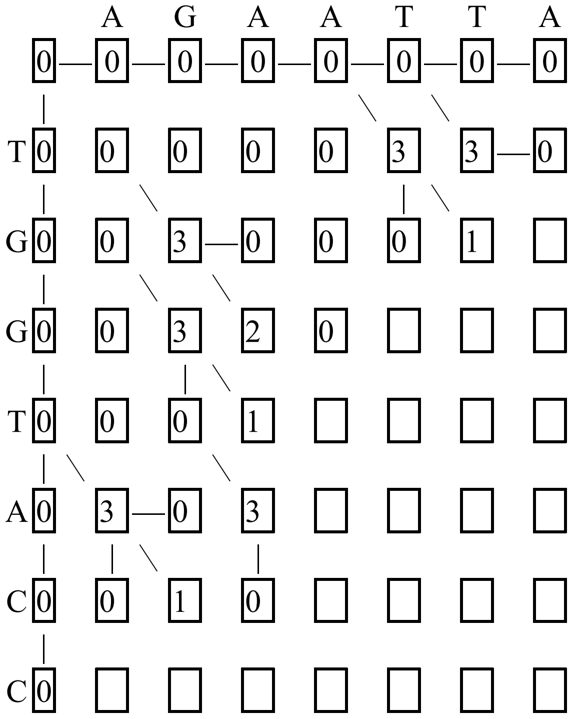
```
                1
             1     1
          1     2     1
       1     3     3     1
    1     4     6     4     1
 1    5    10    10    5     1
1   6   15   20   15   6    1
1   7   21   35   35   21   7   1
```

```
                 1
              1     1
           1     3     1
        1     5     5     1
     1     7    13    7     1
  1     9    25    25    9     1
1    11   41   63   41   11   1
1   13   61   129 129 61   13   1
```

a. What is the rule for generating them?
b. What are the next two rows in each of them?
c. What does the one on the right have to do with string alignment?
d. What is the sum of the numbers in each row?
   1. In the left triangle, you can find an explicit formula.
   2. In the right triangle, you should find a recursive formula. Can you guess what the ratio of successive row-sums approaches
e. Just how far off-topic has this problem gotten?

9. Devise a way to find an optimal local alignment of “ACCACTT” and “TGGTACC” if:
a. Matches are worth +3
b. Mismatches of a purine with a purine, or a pyrimidine with a pyrimidine are worth –1
c. Mismatches of a purine with a pyrimidine are worth –2
d. Gaps are worth –3

Here is a scoring matrix to get you started:



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