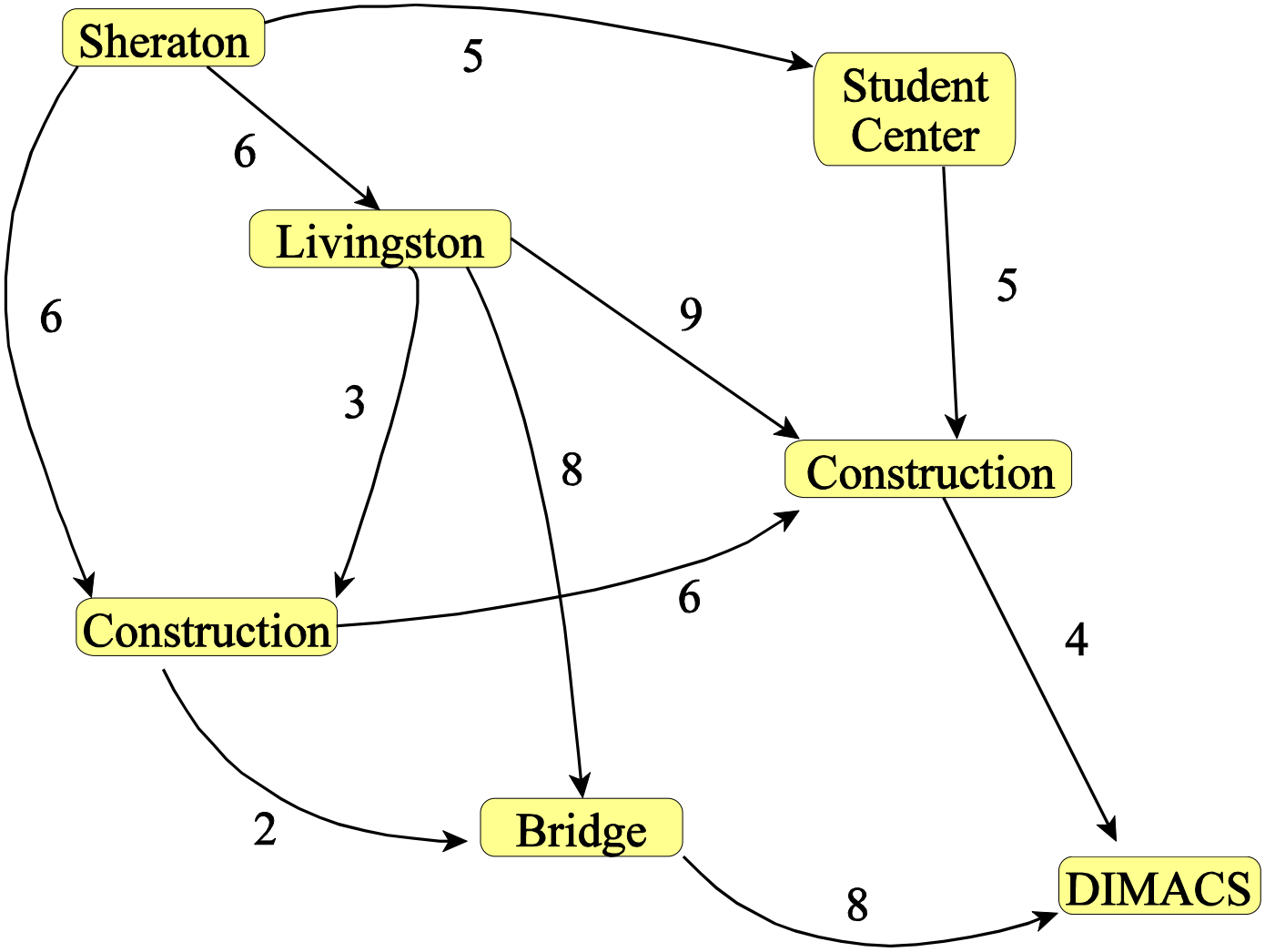
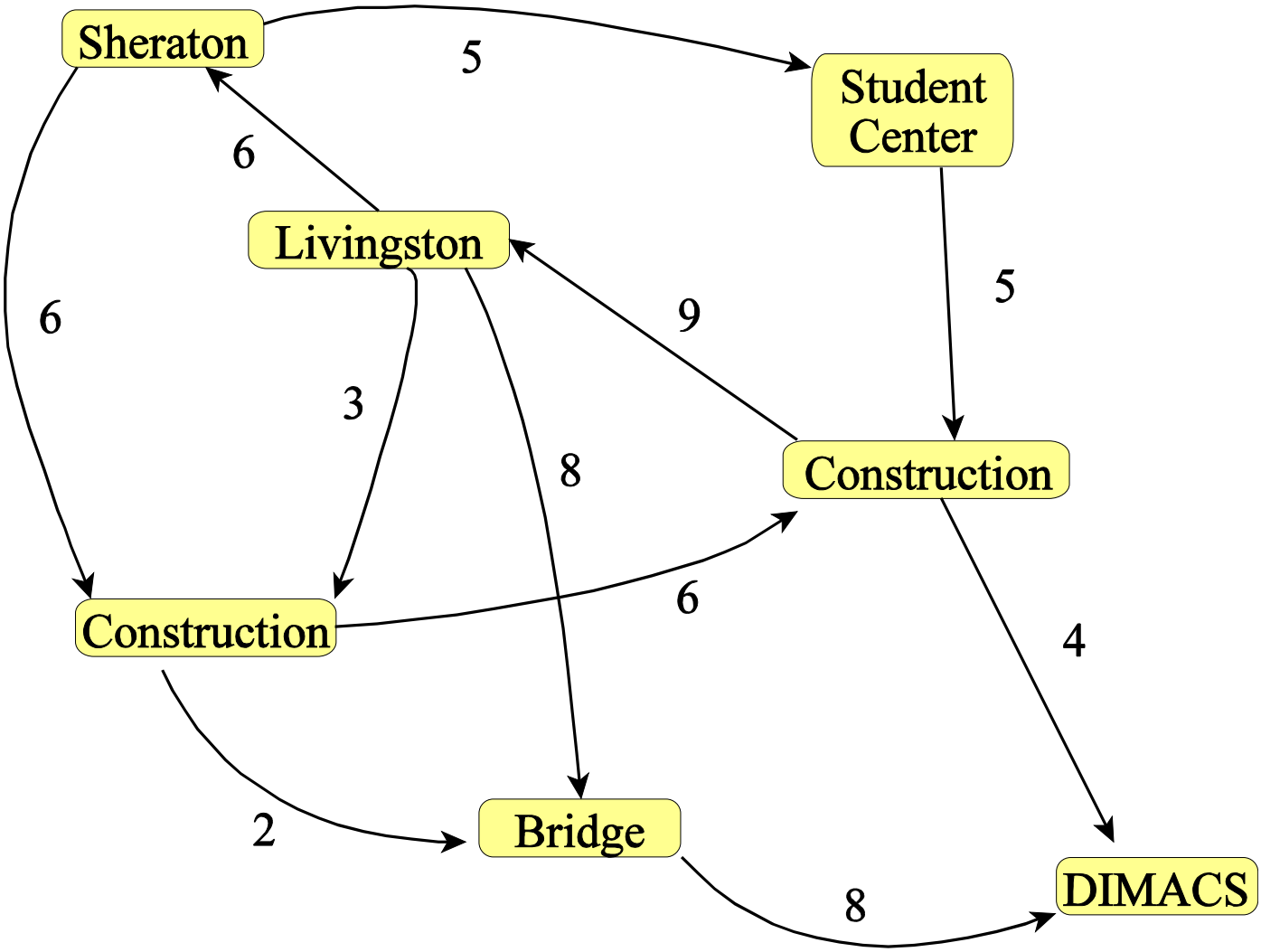
How Did You Get Here this Morning?



- What is the shortest route here from the hotel?

Same Questions, Different Map



- What is the shortest route here from the hotel?

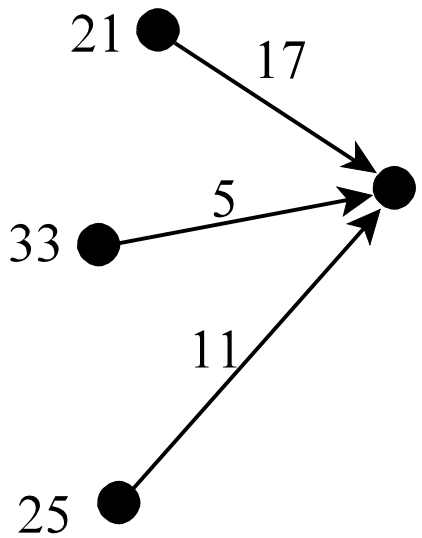
Dijkstra on an Acyclic Directed Graph

Given an acyclic, directed graph with weights on the edges, and some start vertex A, let $d(v)$ denote for each vertex v the length of the shortest path from A to v.

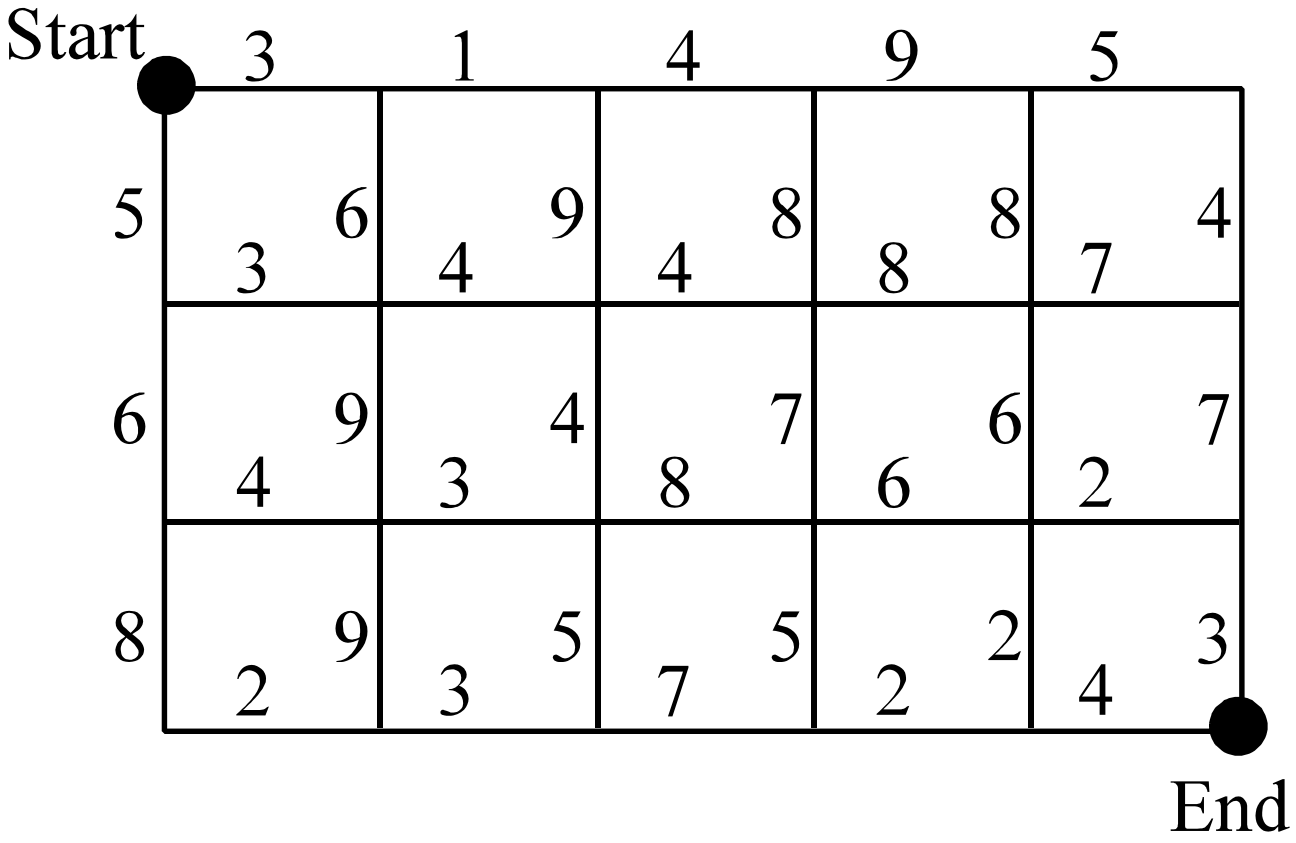
We find all these distances as follows:

- Let $d(\mathrm{A}) = 0$

- For all other vertices v, examine all arcs (u, v) directed into v, compute the sum:
  $$d(u) + \text{weight}(u, v)$$
  and let $d(v)$ be the minimum of these values



A More Typical Setting

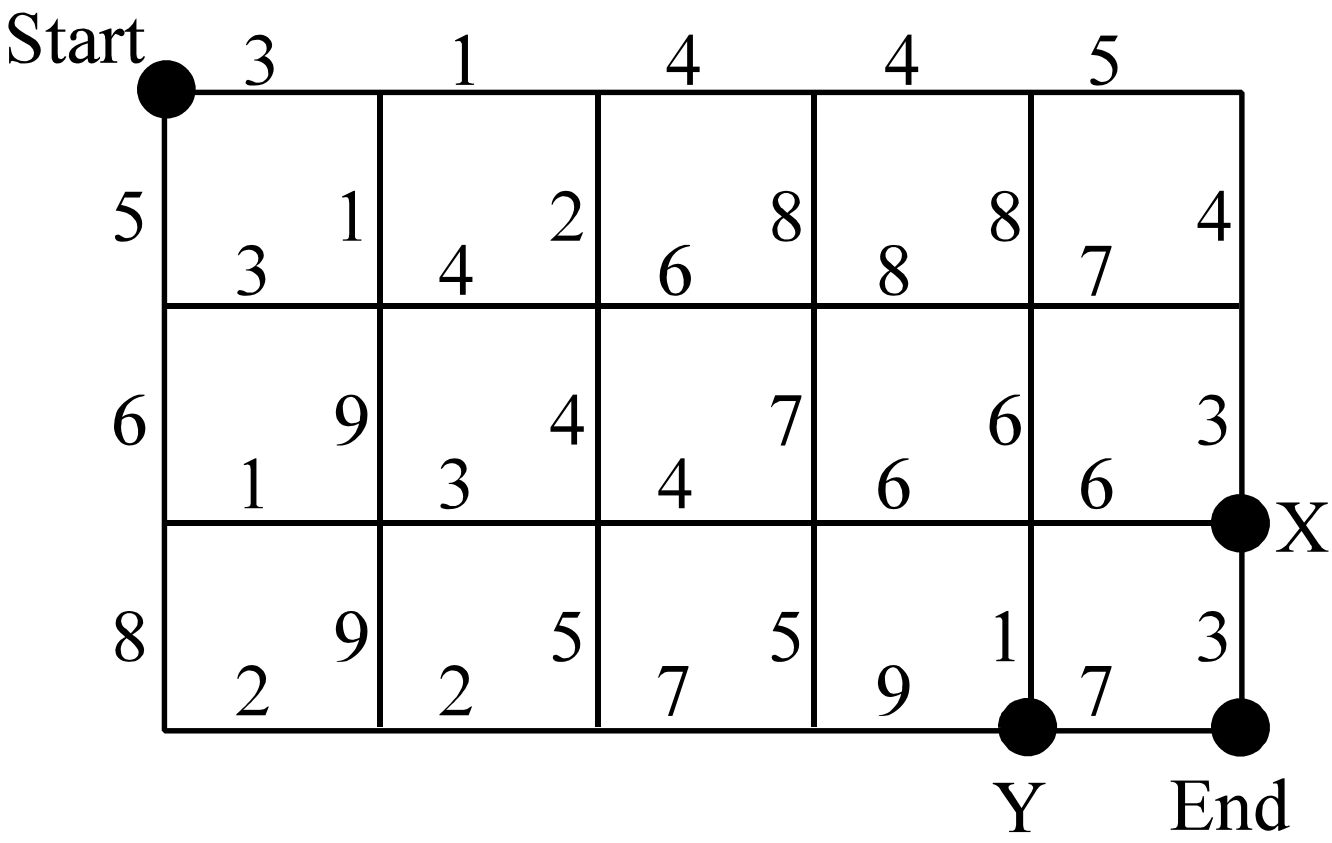


- What is the shortest path from Start to End on this grid, assuming one moves only right or down on each step?

- How many such shortest paths are there?

Dynamic Programming

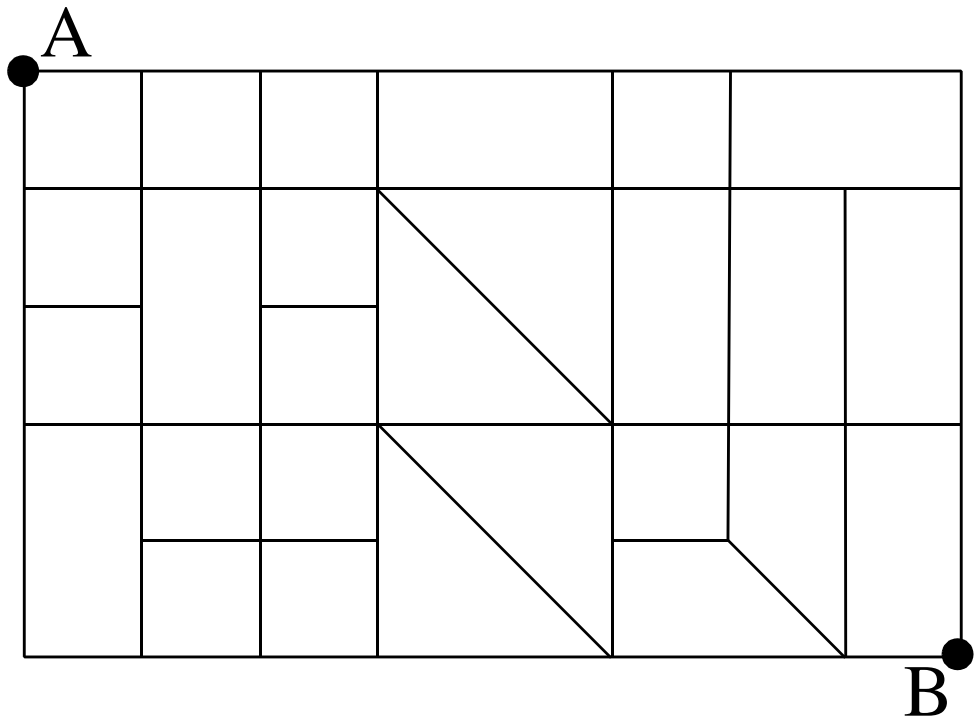
The term "Dynamic programming" refers to the method of finding optimal solutions to a large problem by solving several smaller problems and keeping track of those smaller solutions, usually in order to reuse them.



For example, the shortest path to "End" could be found if we kept track of the shortest paths to "X" and "Y."

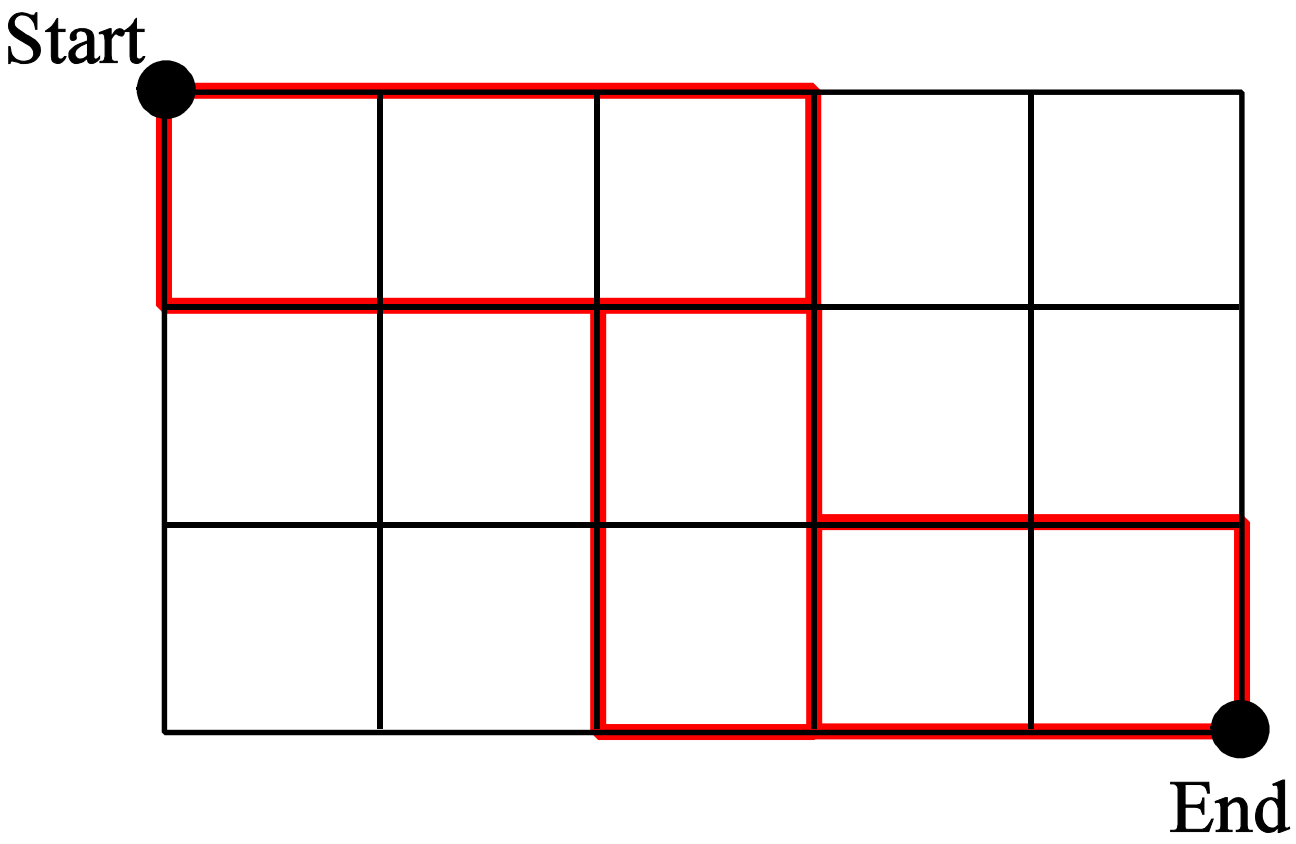
Counting Paths

How many paths are there from A to B which move in a generally South/East direction?



Again, a straightforward dynamic programming solution solves this problem relatively quickly.

Counting Shortest Paths



To enumerate the number of shortest paths, we count the number of ways to walk on our "shortest path backtrack graph" from Start to End.

Global String Alignment

Given two strings, find their best alignment.

Suppose our strings are "AGCGT" and "CAGT." An alignment consists of writing the strings, in order, so that the letters line up in some way. If the strings have different lengths, as is the case here, then we will need to insert some "gap symbol" in the shorter string. We use "−" as our gap symbol.

AGCGT −CAGT	AGCGT C−AGT	AGCGT CA−GT	AGCGT CAG−T
AGCGT CAGT−	AGCG−T C−−AGT	−AGCGT C−−AGT	AGC−GT −−CAGT

Sometimes we will insert gap symbols just to make things line up better, regardless of the lengths.

Which alignment is the best?

A *scoring criterion* is used to evaluate the quality of an alignment.

Gratuitous Use of Gaps

Suppose we wished to align the strings:

GGAGTCCACCTGTGAAACAATA, and
ACGCGCGTCCTCCTGTGACAATT

Notice here how the insertion of some gaps can help us get a good alignment of a fairly lengthy region of similarity:

GGA---GTCCACCTGTGAAACAATA
ACGCGCGTCCTCCTGTGA--CAATT

We will thus allow for insertion of gaps wherever it will help us to align our sequences more optimally.

Scoring Criterion

AGCGT −CAGT	AGCGT C−AGT	AGCGT CA−GT	AGCGT CAG−T
AGCGT CAGT−	AGCG−T C−−AGT	−AGCGT C−−AGT	AGC−GT −−CAGT

Look at the columns of the alignmnet. Score:

+2 For each alignment of matched letters
–1 For each alignment of mismatched letters
–2 For each alignment of a letter with a gap
$-500 For each alignment of two gaps

Given this scoring system, how do we find the optimal alignment. How do we find all optimal alignments?

Are There Many Possible Alignments?

We wish to align ABCDEFG with TUVWXYZ.

Our first column can align either:

- A with T
- A with a gap
- a gap with T

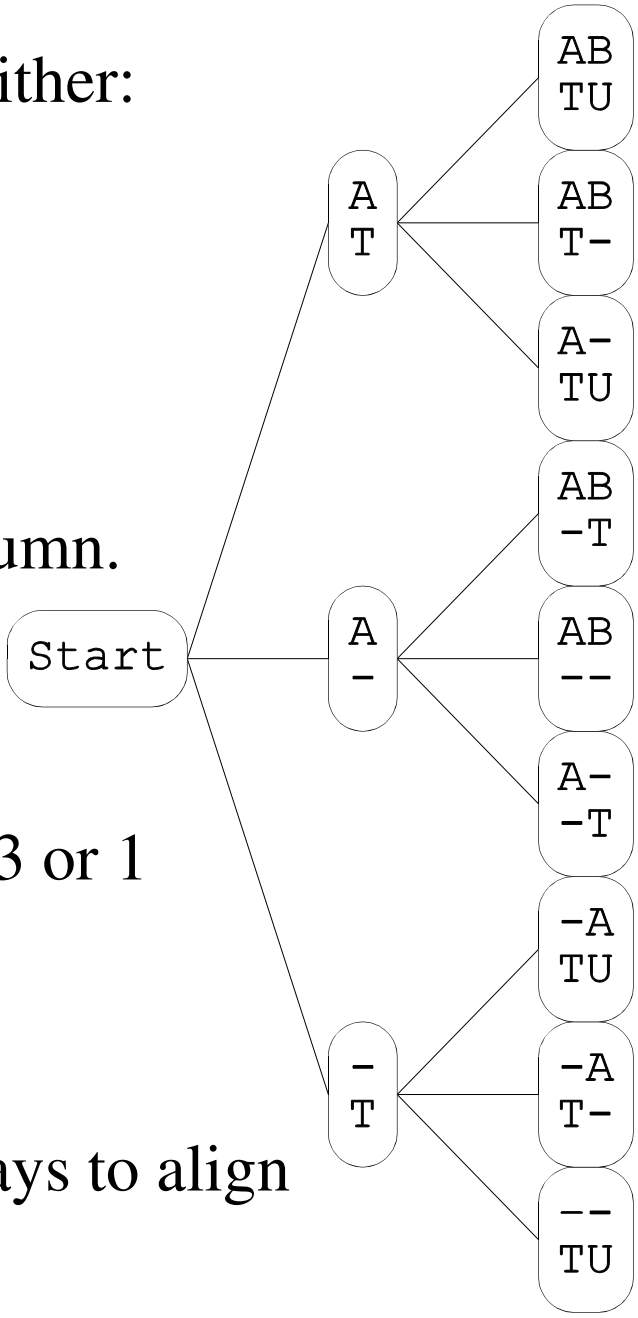
for 3 possible choices.

Similarly for the second column.

And the 3rd through 7th.

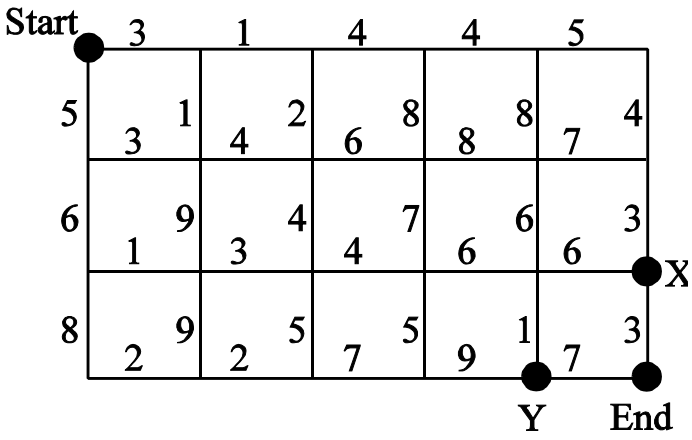
The 8th column could have 3 or 1 choices, depending on what happened before.

Thus, there are at least 3^7 ways to align these strings.



In general, if both strings are of length at least k, then there are at least 3^k ways to align them.

Dynamic Programming to the Rescue



Recall what we said about the figure to the right: If we knew the best paths to X and Y, we could easily find the best path to End.
What information would be analogously helpful in attempting to find an optimal alignment of AGCGT and CAGT?

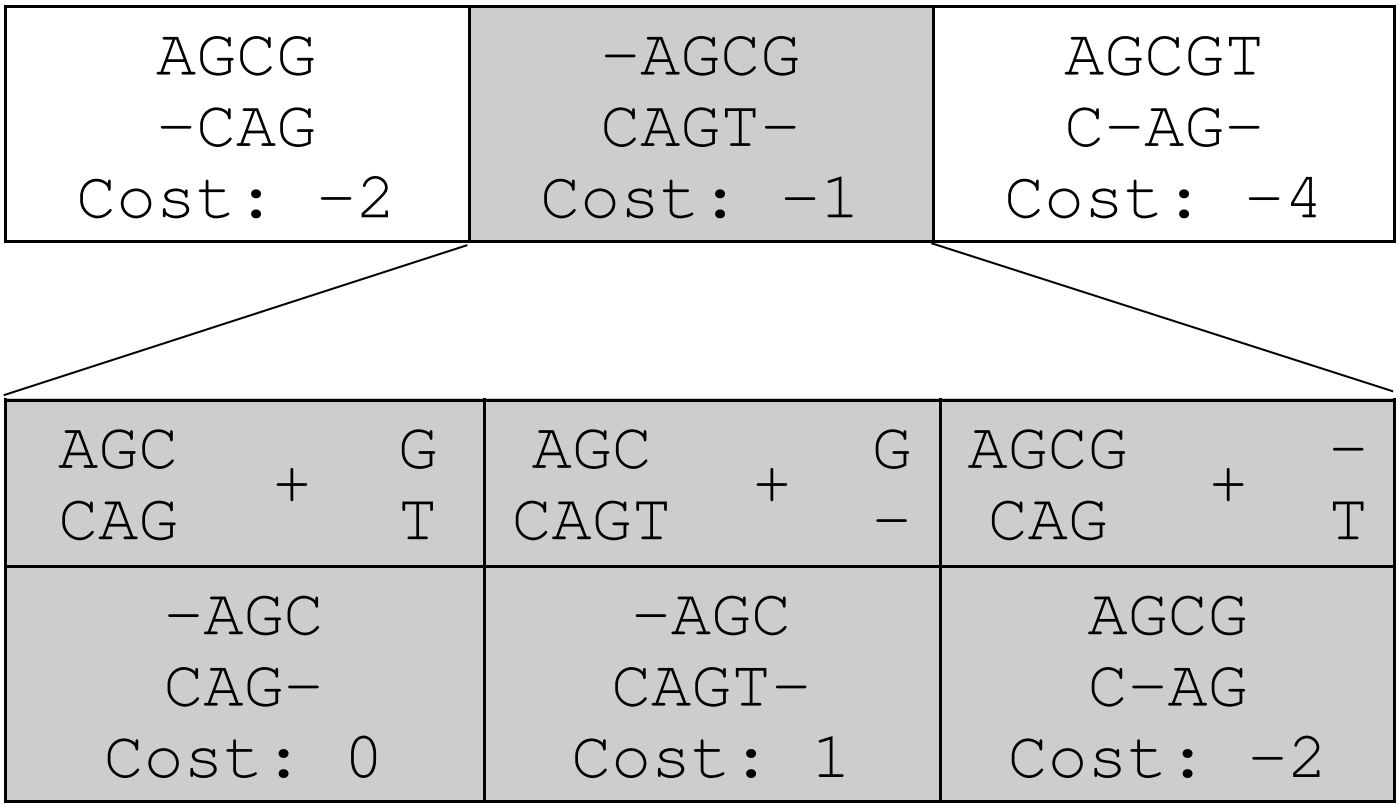
What could our last column look like?

Either $\begin{bmatrix} T \\ T \end{bmatrix}$, $\begin{bmatrix} T \\ - \end{bmatrix}$ or $\begin{bmatrix} - \\ T \end{bmatrix}$.

So what optimal alignments would you like to know?

AGCG + T CAG + T	AGCG + T CAGT + -	AGCGT + - CAG + T
AGCG -CAG Cost: -2	-AGCG CAGT- Cost: -1	AGCGT C-AG- Cost: -4

How Did We Get Those Costs?

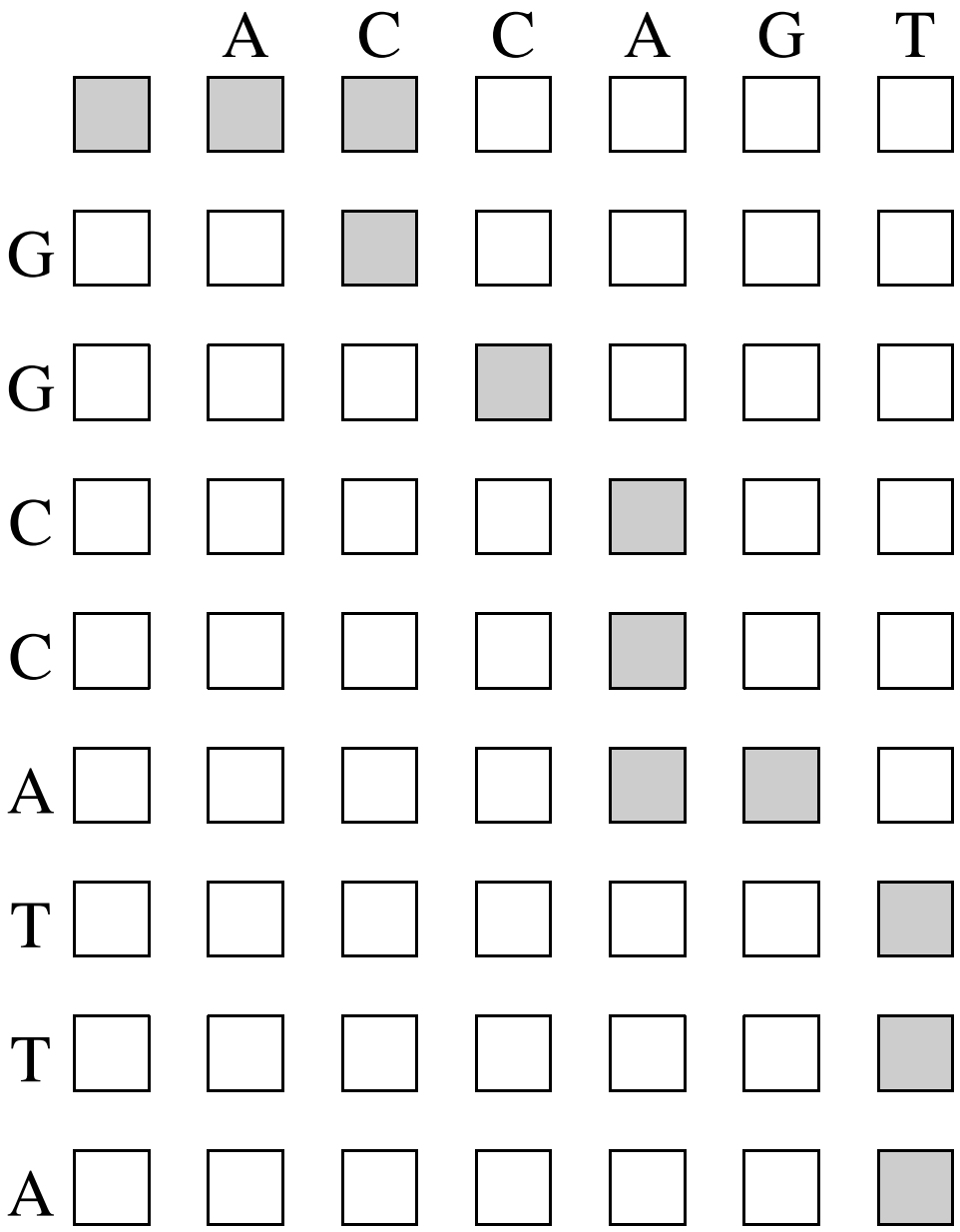


This suggests that, as in the case of our Dijkstra's algorithm, we can "reuse" our solutions to smaller sub-problems to solve the larger problems.

The idea:
Find an optimal alignment between *each* pair of prefixes for each of the two strings.

ACGAAGACTTTTTACA
GGCTACGATACGA

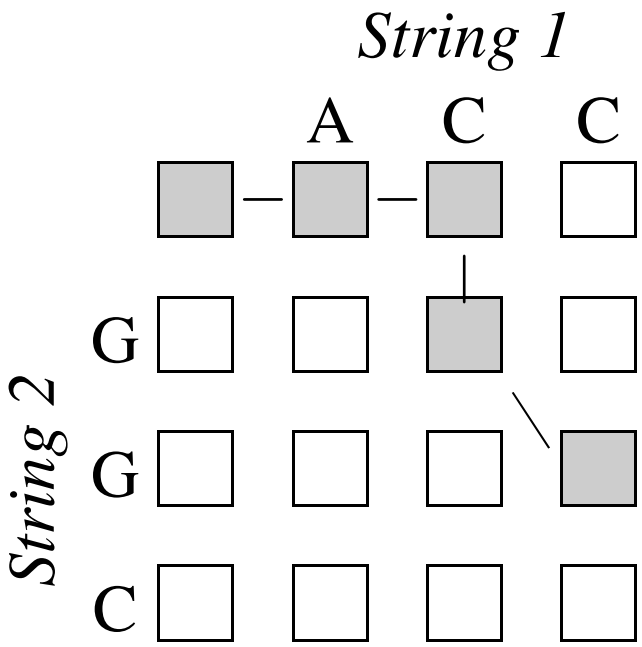
A Nifty Picture



This diagram corresponds to the alignment:

```
AC-CA--GT--
--GGCCA-TTA
```

The Correspondence



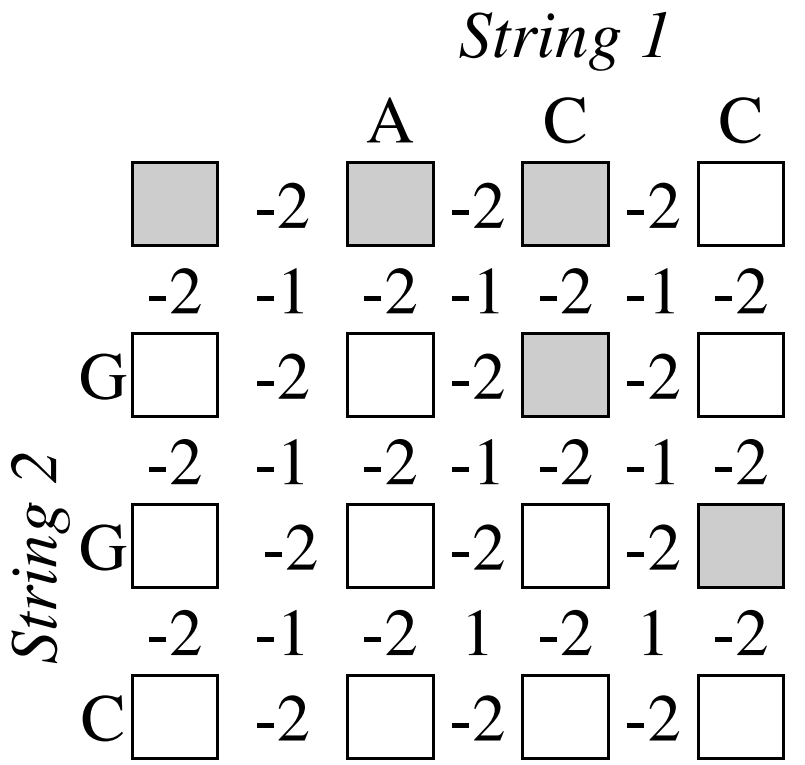
- Each column of the alignment corresponds to a transition from one shaded box to the next, having the letters of the new row and/or new column

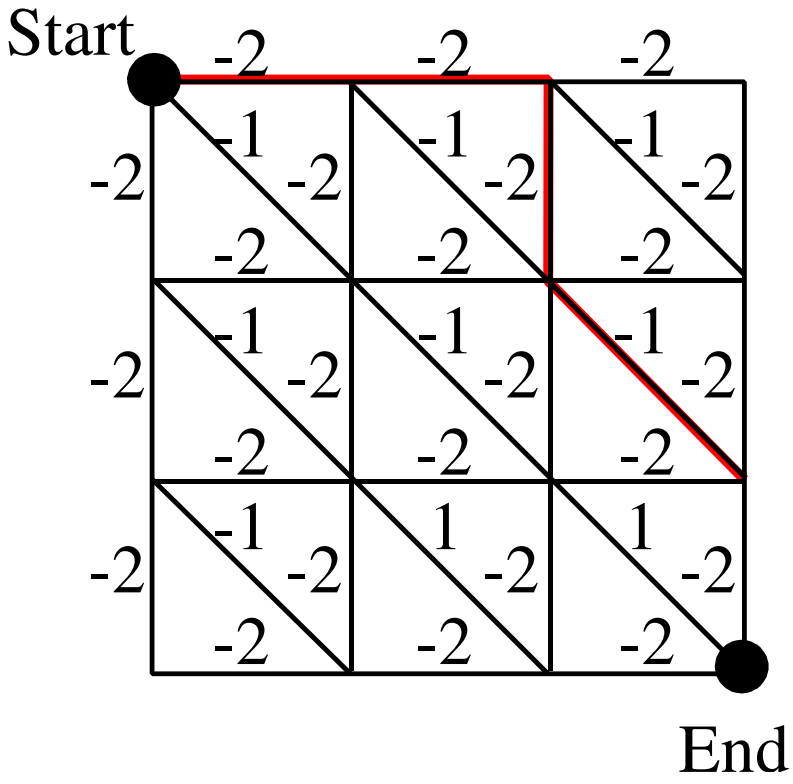
- Moving horizontally corresponds to aligning a letter in string 1 with a gap in string 2

- Moving vertically corresponds to aligning a letter in string 2 with a gap in string 1

- Moving diagonally corresponds to aligning the two characters from string 1 and string 2

The Cost of Each Step



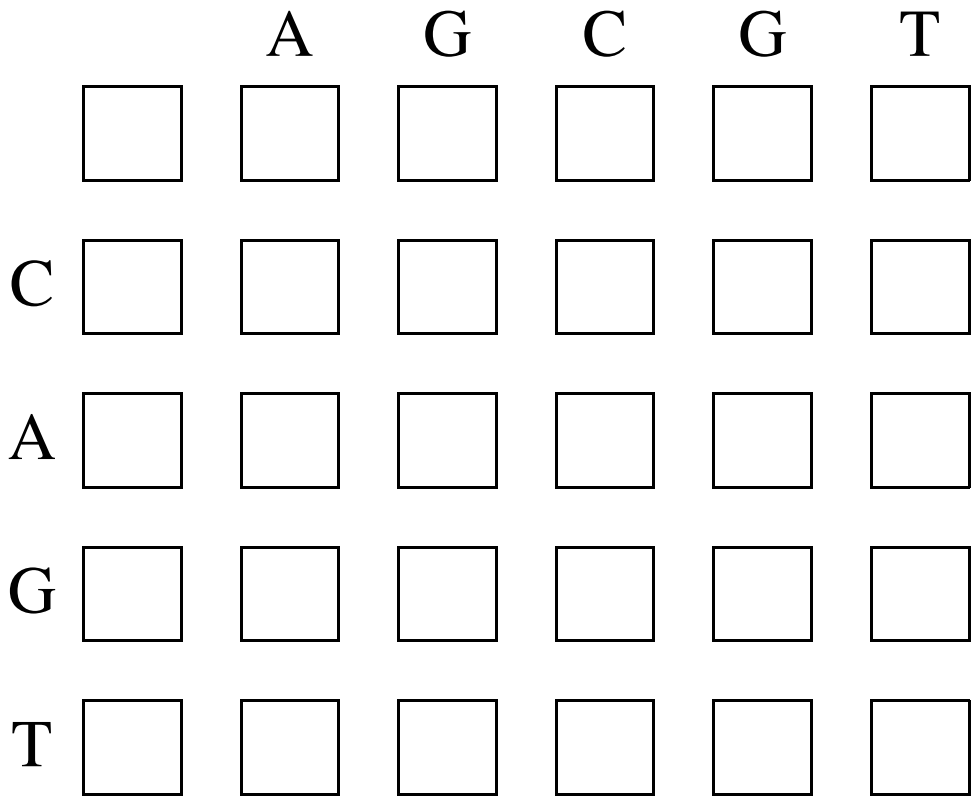


An Optimal Alignment is a Best Path

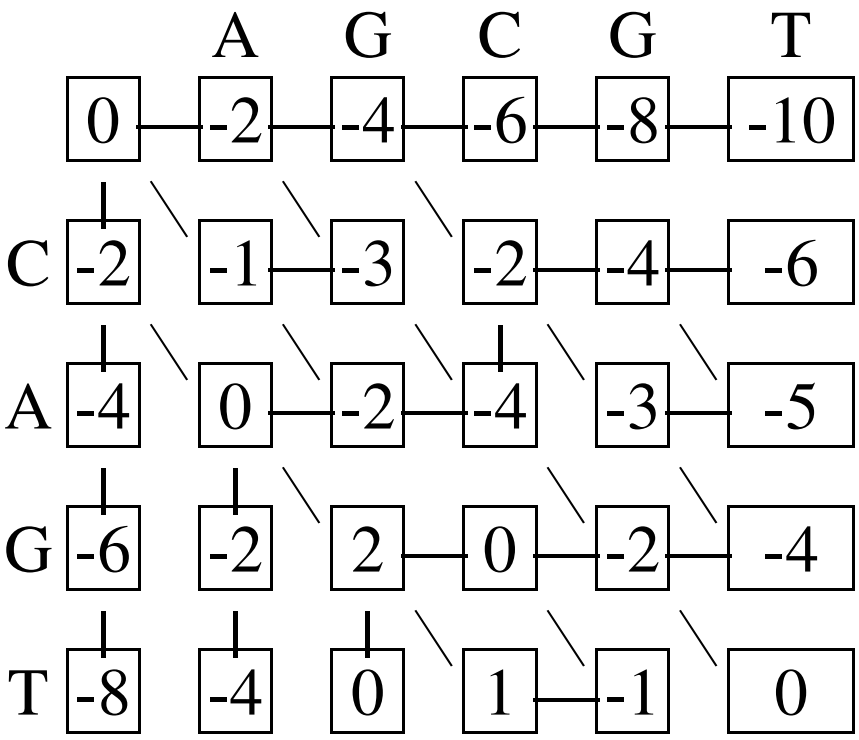
So we can find an optimal alignment by finding the best path in this graph

Which can be done by dynamic programming, determining the "distance" to each cell by considering the distances to its neighbors. Let's try this together:

Match	+2
Mismatch	–1
Gap	–2

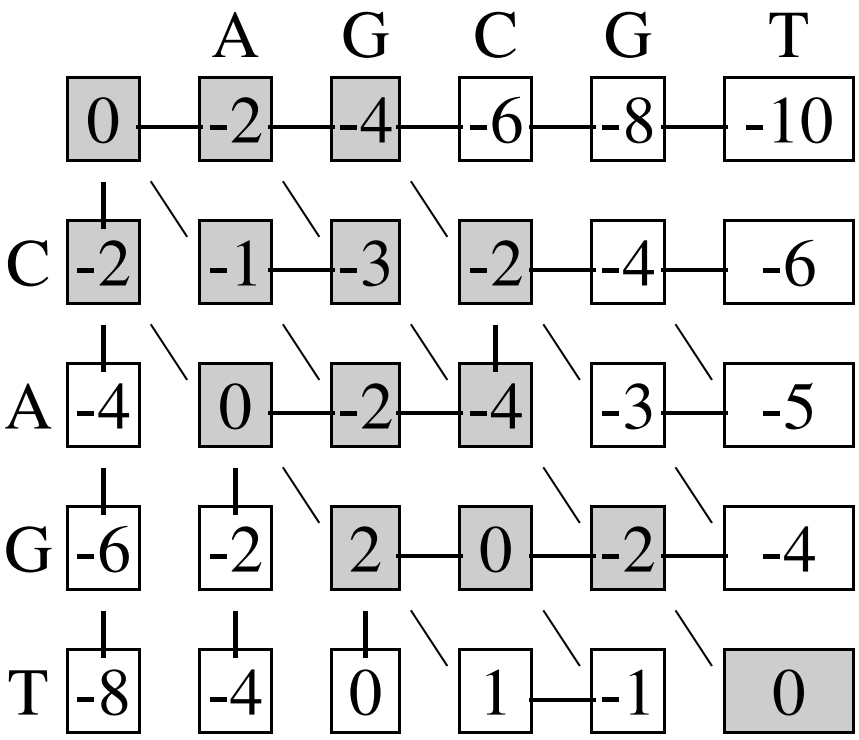


Reconstructing the Alignment



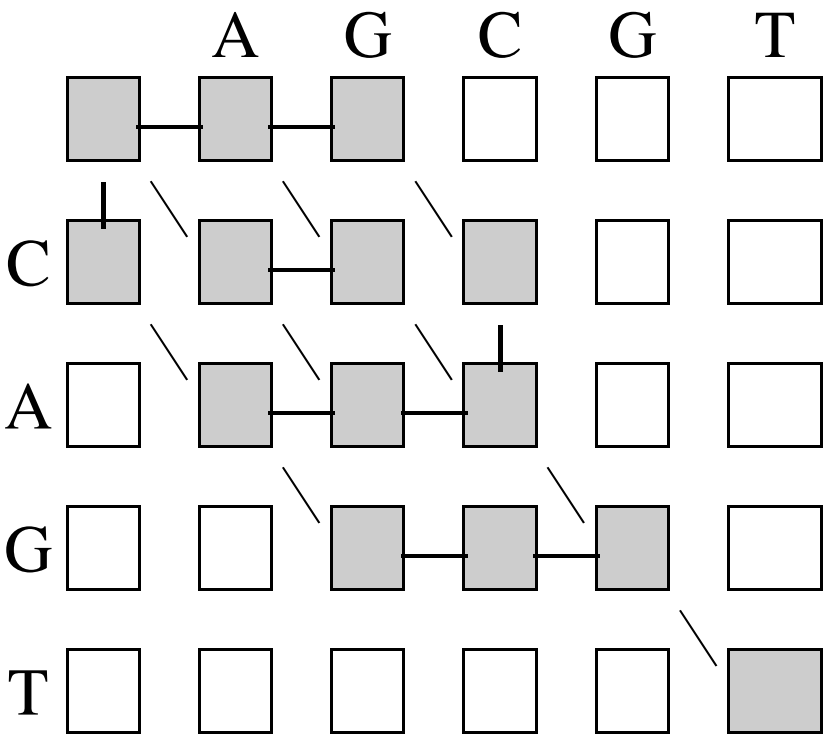
When each cell is optimized, record all (1, 2 or 3) cells from which that optimum can be achieved.

Reconstructing the Alignment

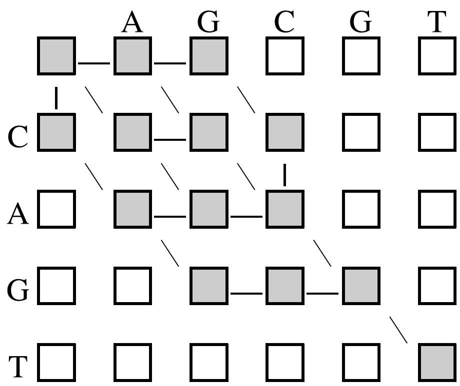


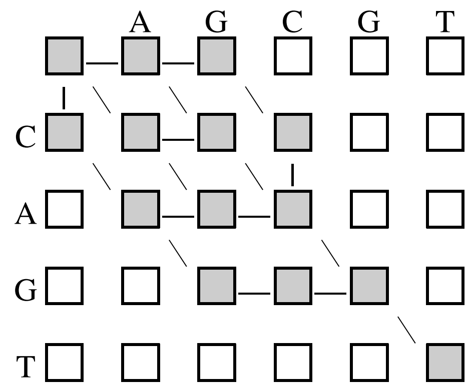
- Shade the bottom-right cell
- For each other cell, shade it if there is a directed edge *from* that cell *to* a shaded cell
- These shaded cells form some subgraph of the original lattice
- (These should be figured bottom-to-top, right-to-left)
- All paths in this subgraph correspond to optimal alignments

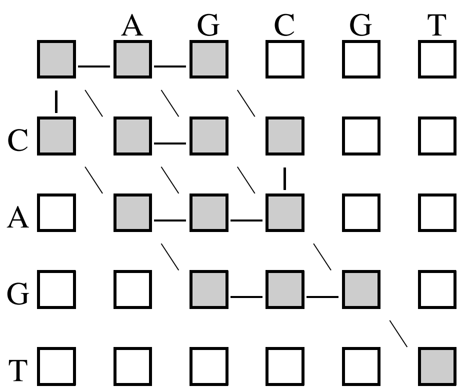
Counting the Optimal Alignments

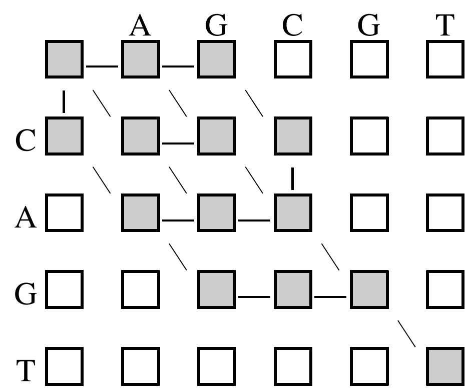


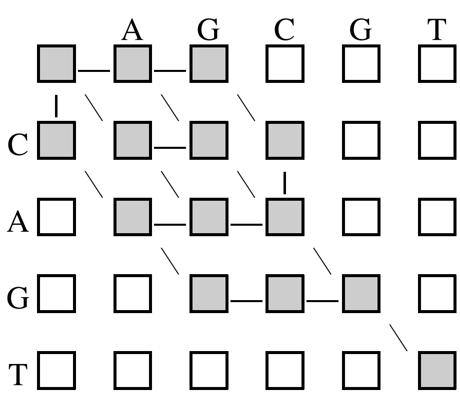
Building the Optimal Alignments

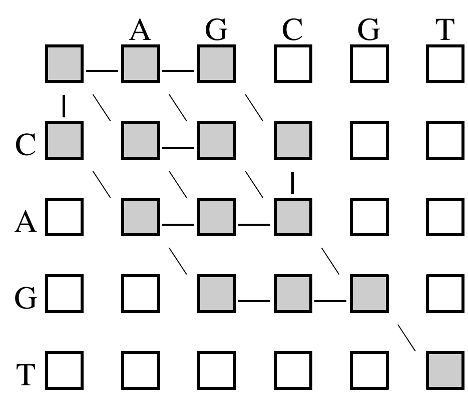












Semi-Global Alignment

What if:

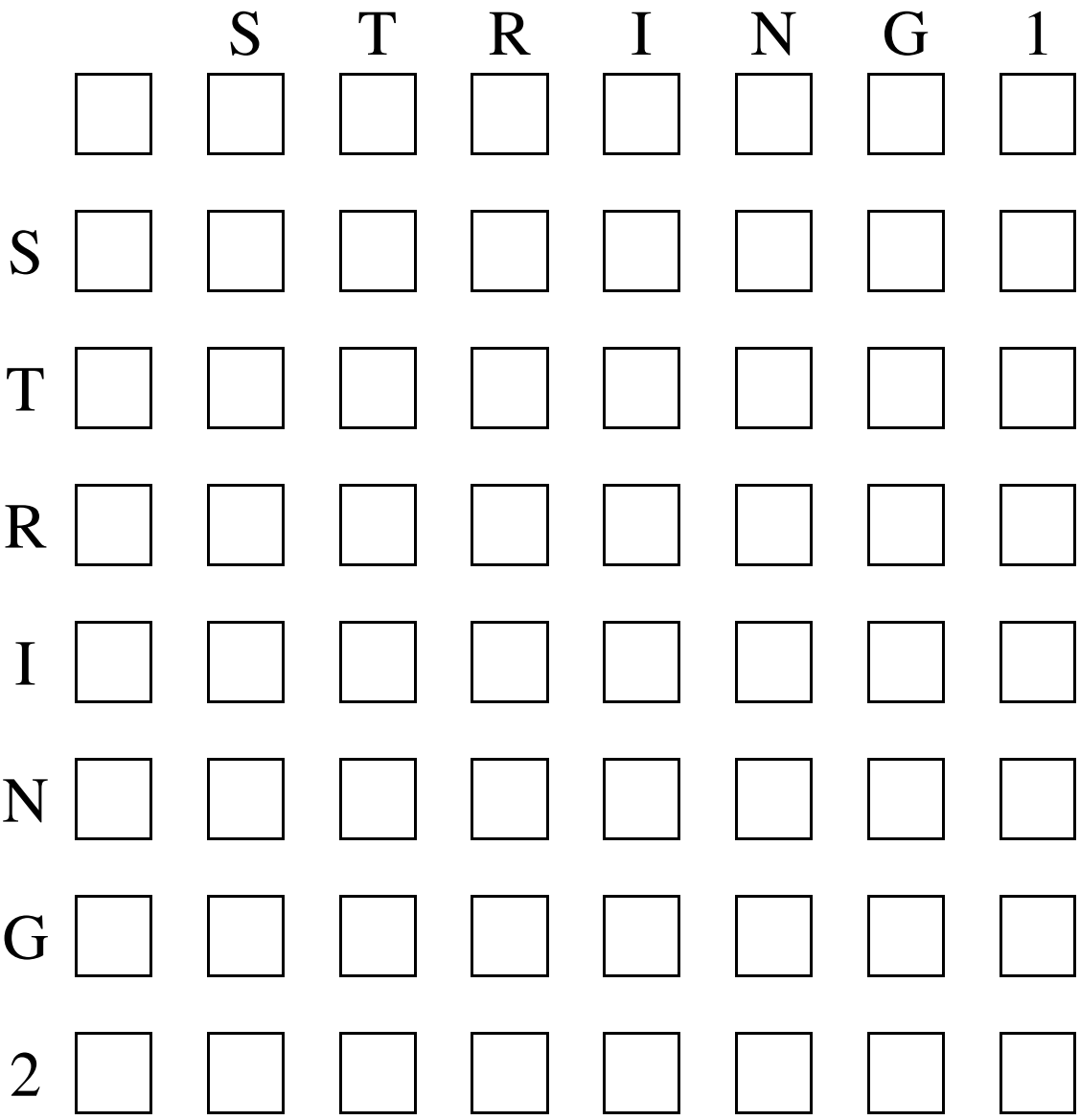
- Gaps were not penalized at the start of string 1
- Gaps were not penalized at the start of string 2
- Gaps were not penalized at the end of string 1
- Gaps were not penalized at the end of string 2
- Any combination of the above?

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?

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

How do we reflect this in our dynamic program?

Semi-Global Alignment



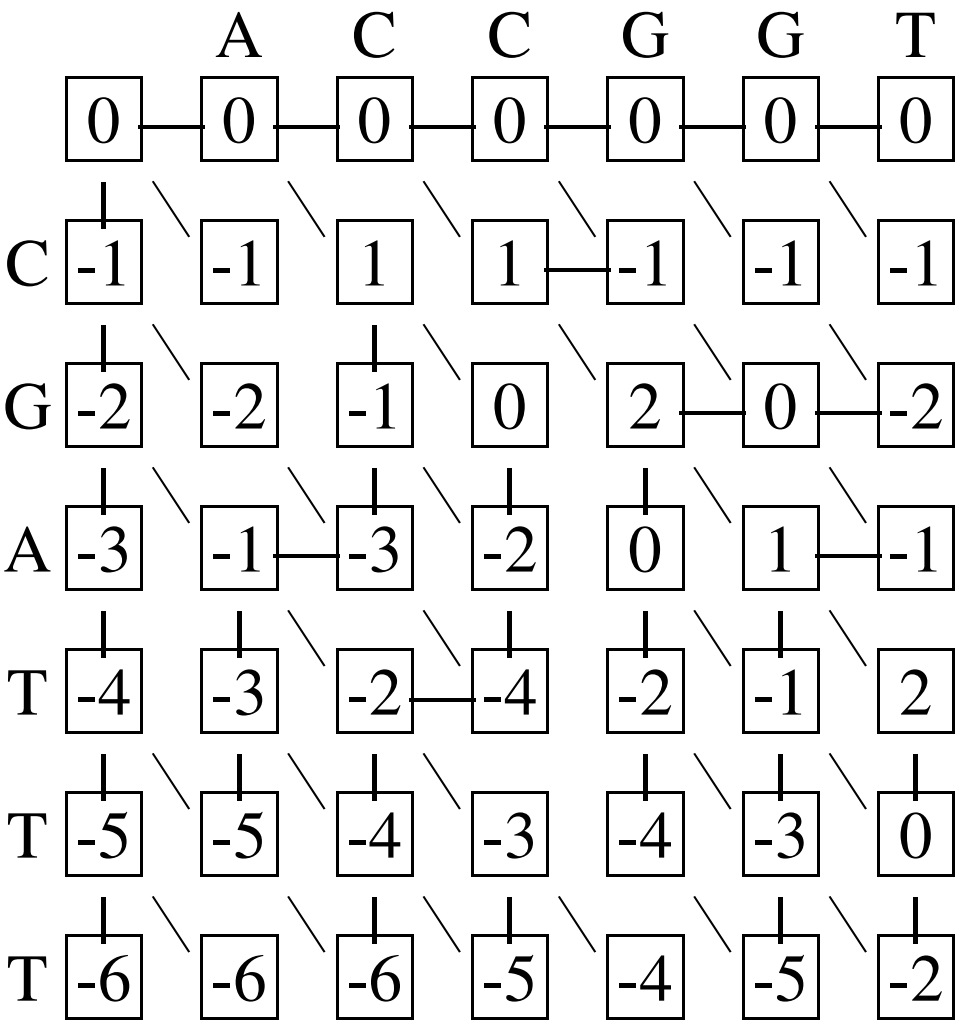
What would be different about our computation if we did not charge for gaps at the beginning or end of one string or another?

Semi-Global Alignment

		S	T	R	I	N	G	1
	0	0	0	0	0	0	0	0
S		*For free initial gaps in string 2, initialize this row to all "0"s*						
T								
R								
I								
N								
G		*For free end gaps in string 2, select the greatest element in the last row, and align accordingly*						
2	-7	-8	-3	-4	-5	-4	-5	-6

And similarly for dealing with string 1.

Guess my Gap Penalties



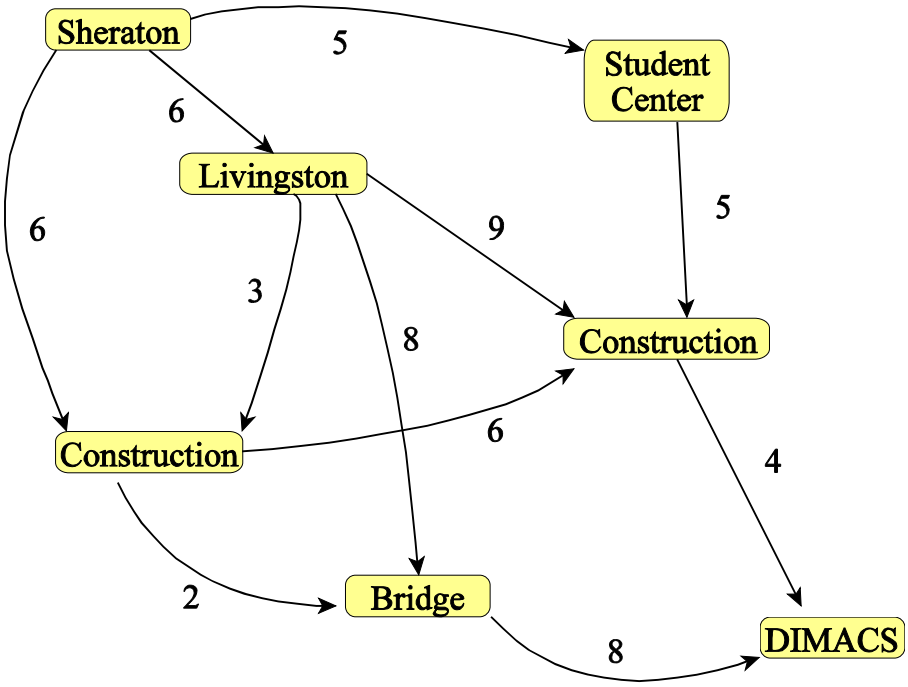
String1 initial gap penalty: –1
String2 initial gap penalty: 0

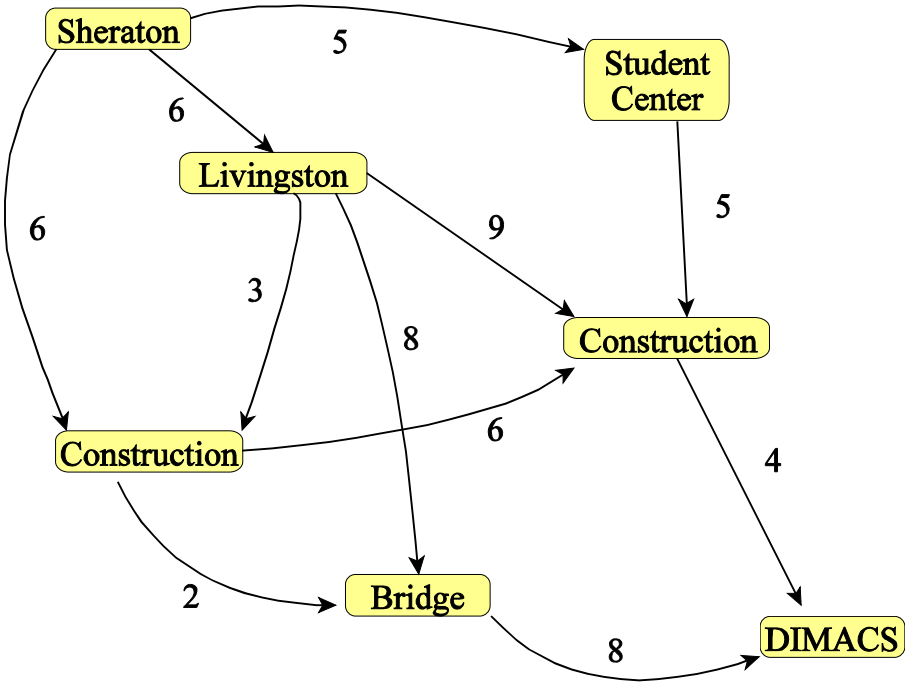
Internal gap penalty: –2

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

Handout #1 — Shortest Path from the Hotel

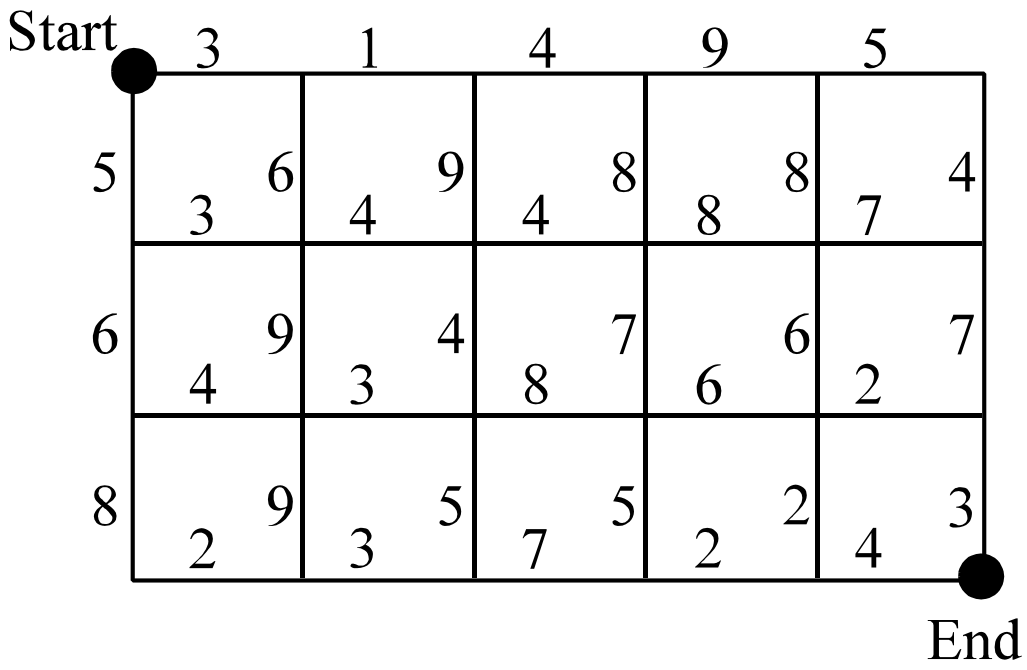
What is the length of the shortest path from the hotel to DIMACS? Use whatever method you wish to find the answer. (Two copies of the graph are provided.)

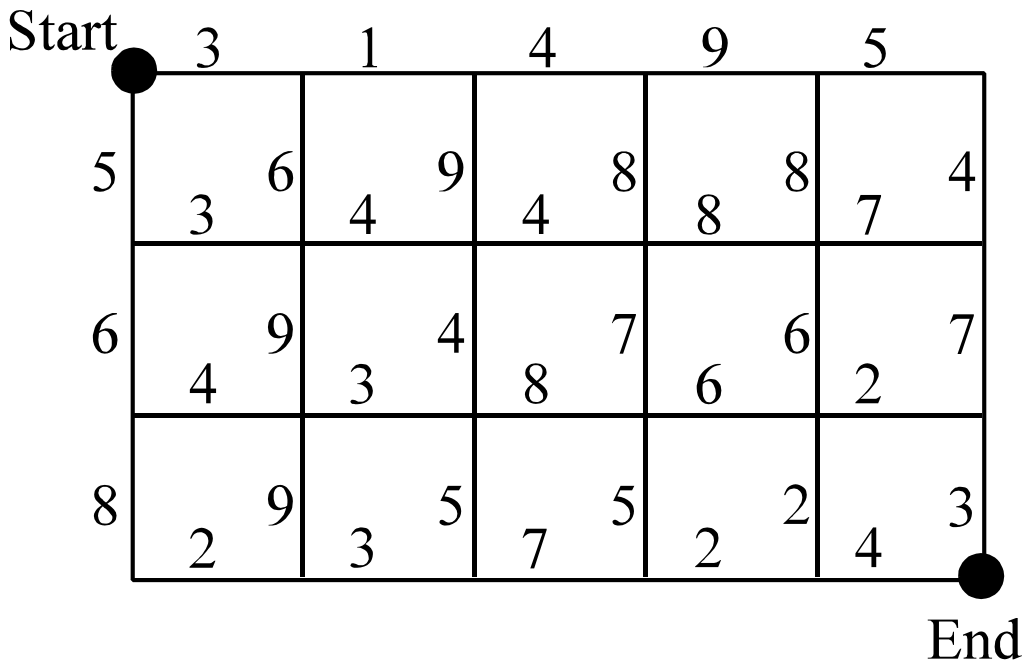




Handout #2 — Directed Acyclic Dijkstra

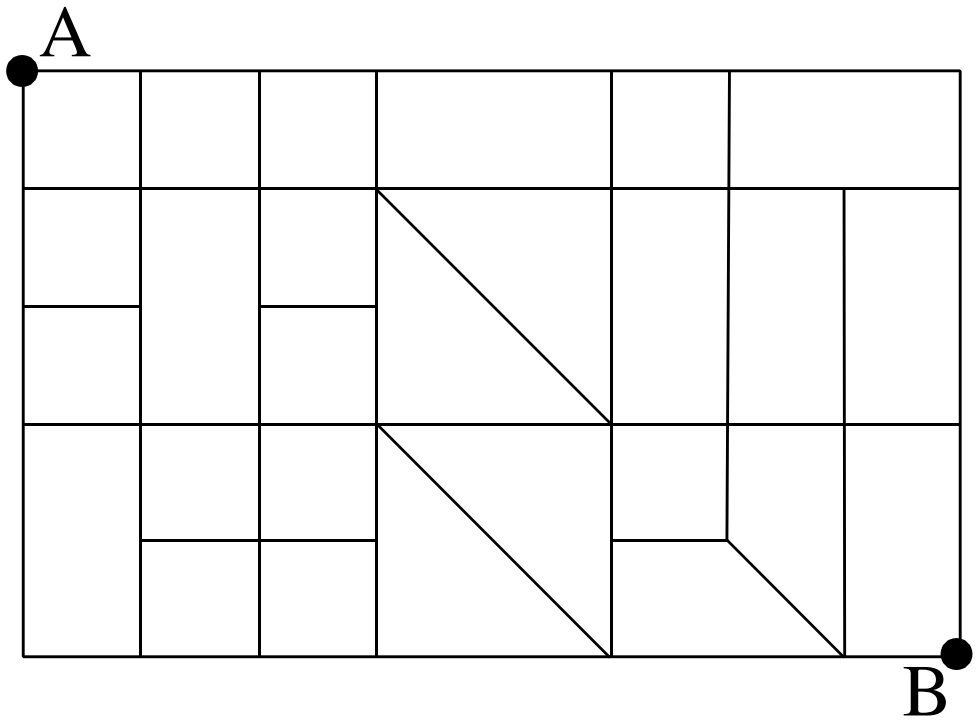
Find a shortest path from Start to End on this graph. Two copies are provided.

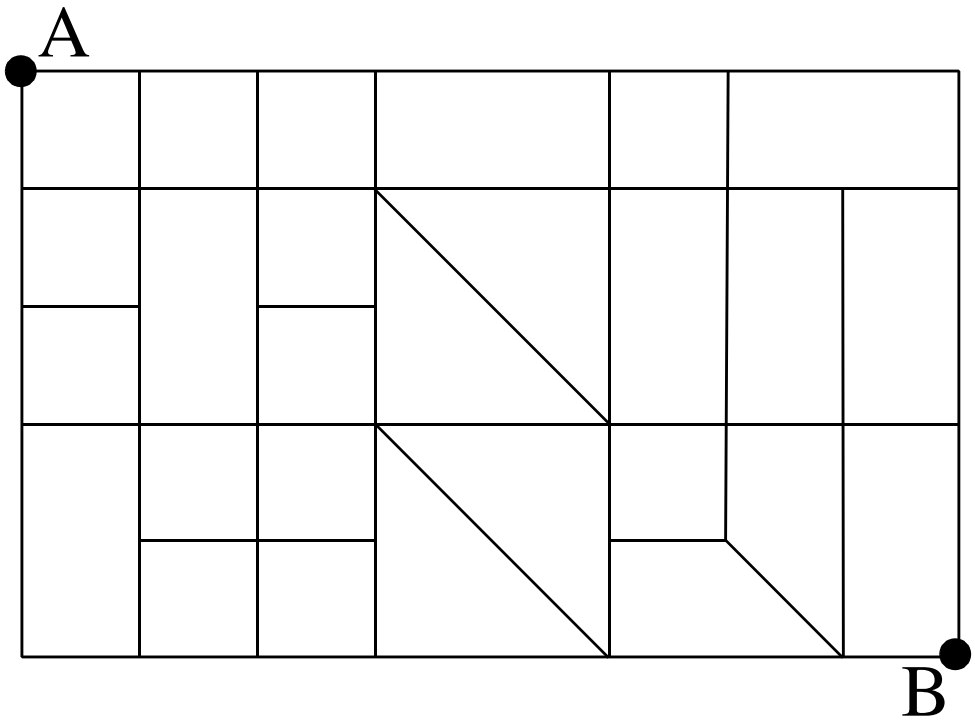




Handout #3 — Counting Paths

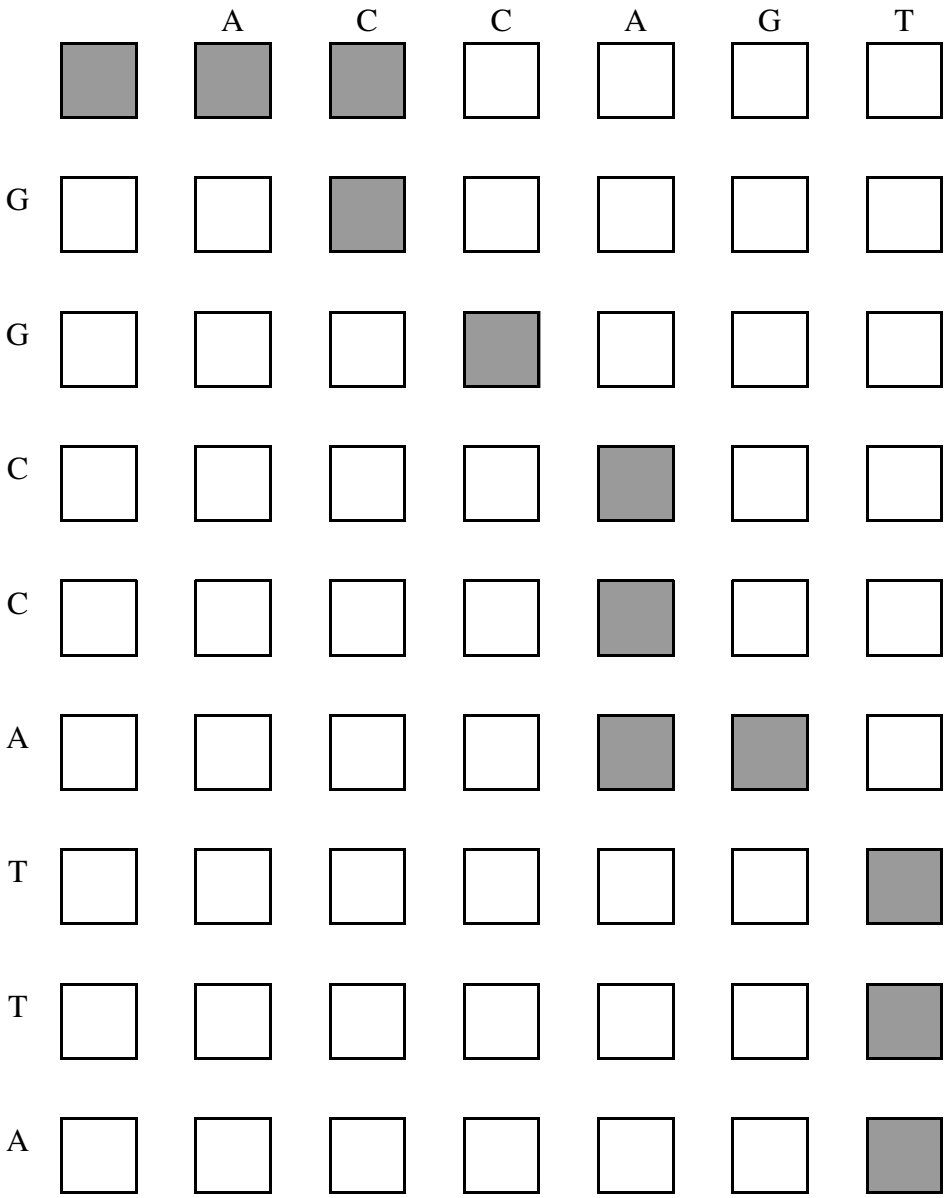
How many ways are there to walk from A to B on this graph, if you are not allowed to "backtrack." That is, all steps should be toward either the East, the South or the Southeast.





Handout #4 — The Correspondence Between Paths and Alignments

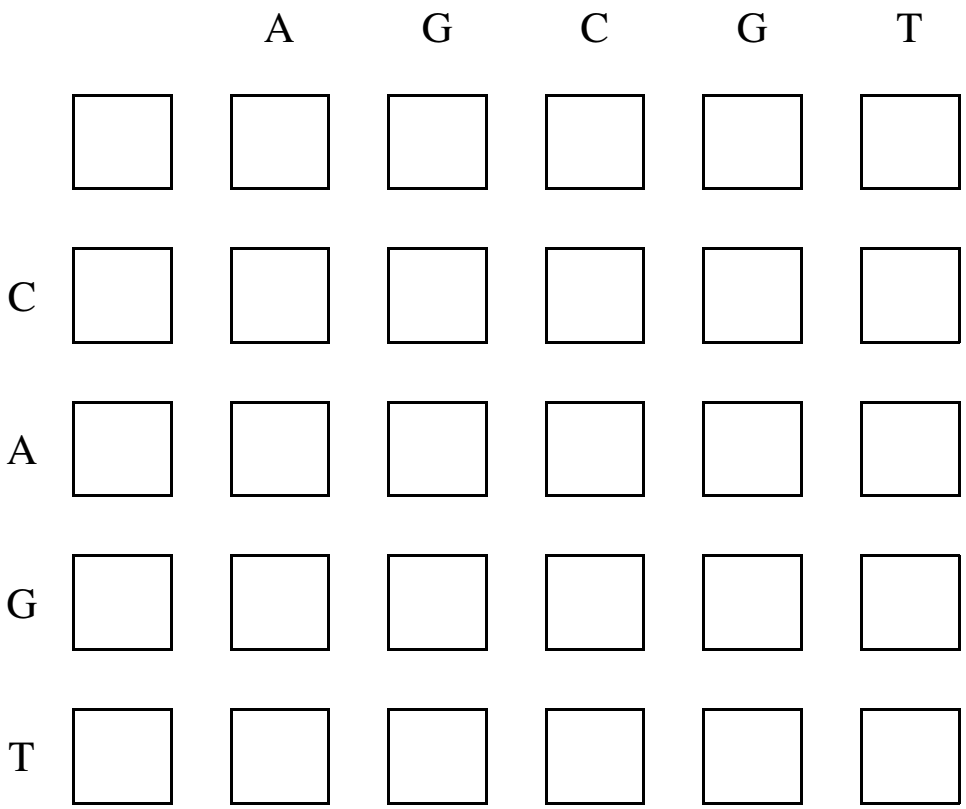
What alignment is indicated by the shaded boxes below?



Handout #5 — Finding the Optimal Alignment via Dynamic Programming

Use dynamic programming to find the best alignment (that with the highest score) between the two indicated strings.

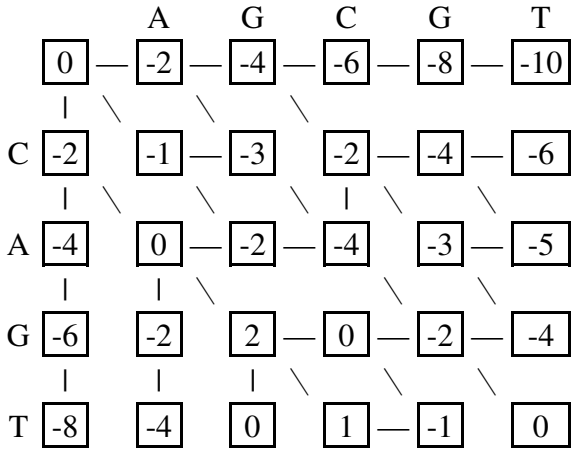
Match	+2
Mismatch	−1
Gap	−2

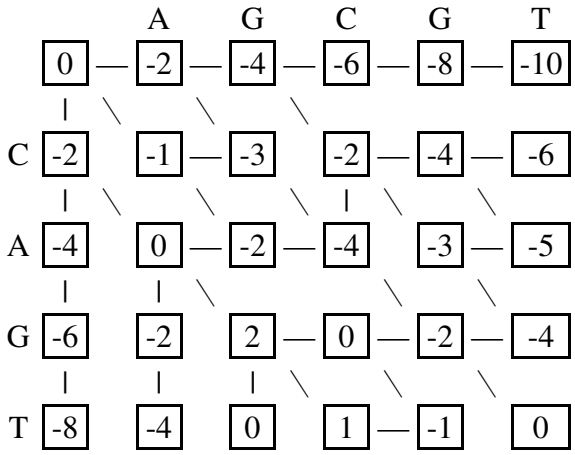


Handout #6 — Enumerating the Optimal Alignments from our Table

Here is the completed dynamic programming matrix for these two strings. (Two copies are proided.) How many optimal alignments are there? What are all of the optimal alignments?

Match	+2
Mismatch	−1
Gap	−2

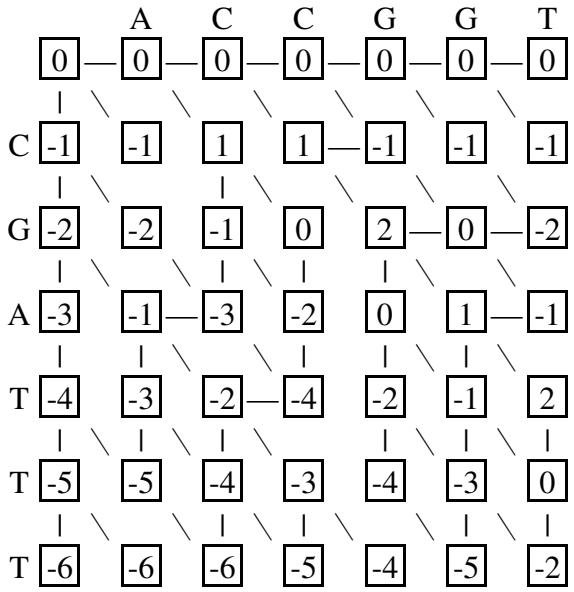


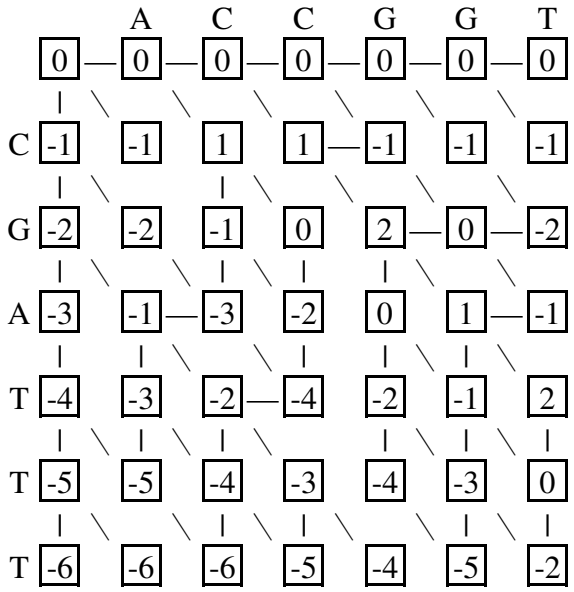


Handout #7 — Varying the Gap Penalties

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 optiaml score if gaps are penalized at the end of the strings?

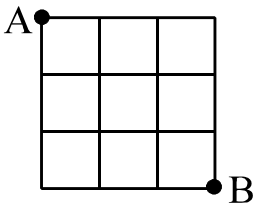




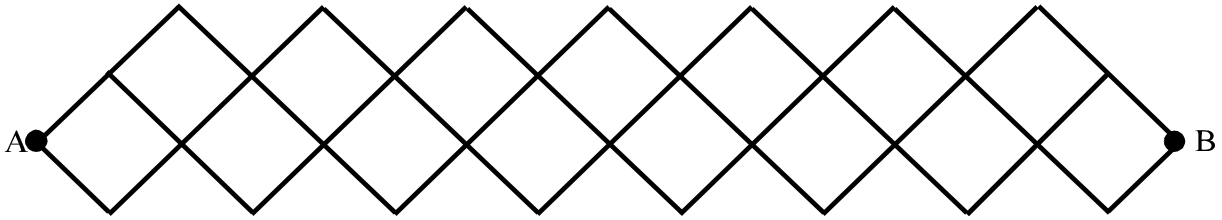
Exercises — Introduction to Dynamic Programming

Quick Concepts

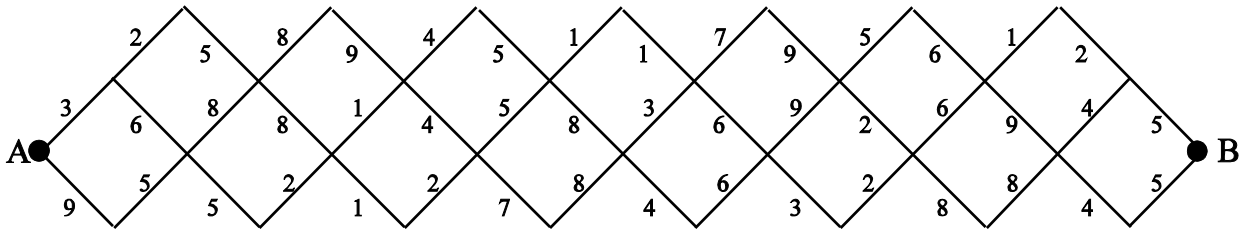
1. How many ways are there to walk from A to B on the grid to the right, without backtracking?



2. Same question on the grid below.



3. The numbers on the edges of the graph below represent distances. What is the length of the shortest path from A to B? How many routes achieve that length?



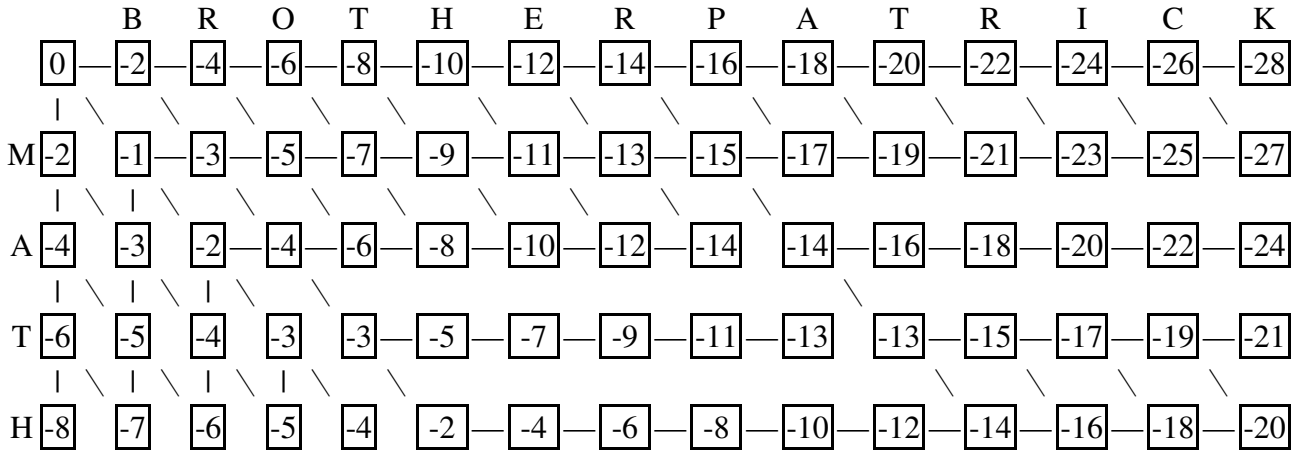
4. What alignment corresponds to the following scoring matrix:

		C	C	A	G	G	T	A
	0	-2	-4	-6	-8	-10	-12	-14
A	-2	-1	-3	-3	-5	-7	-9	-11
G	-4	-3	-2	-4	-2	-4	-6	-8
G	-6	-5	-4	-3	-3	-1	-3	-5
A	-8	-7	-6	-3	-4	-3	-2	-2
A	-10	-9	-8	-5	-4	-5	-4	-1

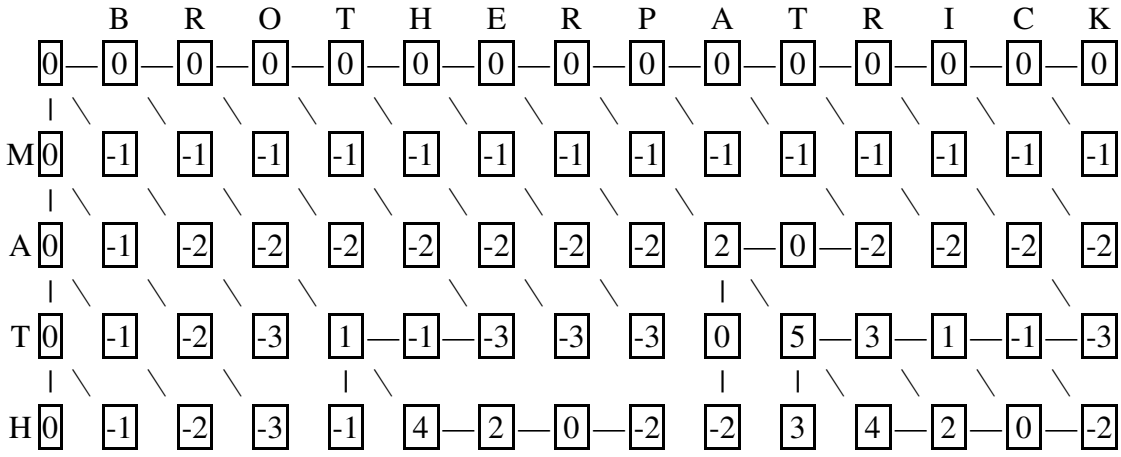
Copyright 2004 — DIMACS BioMath Connect Institute — Robert Hochberg.

Presentation Problems

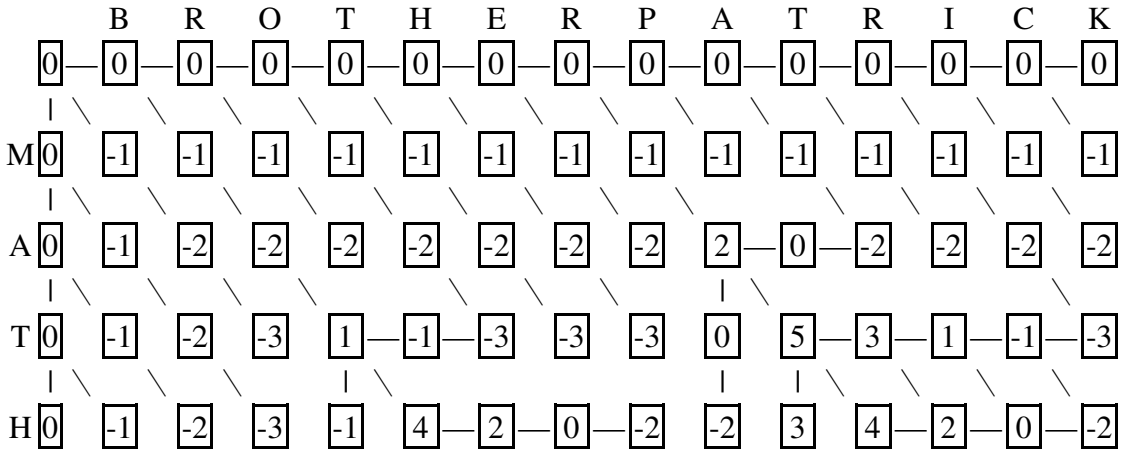
5. How many optimal alignments are indicated by the following scoring matrix?

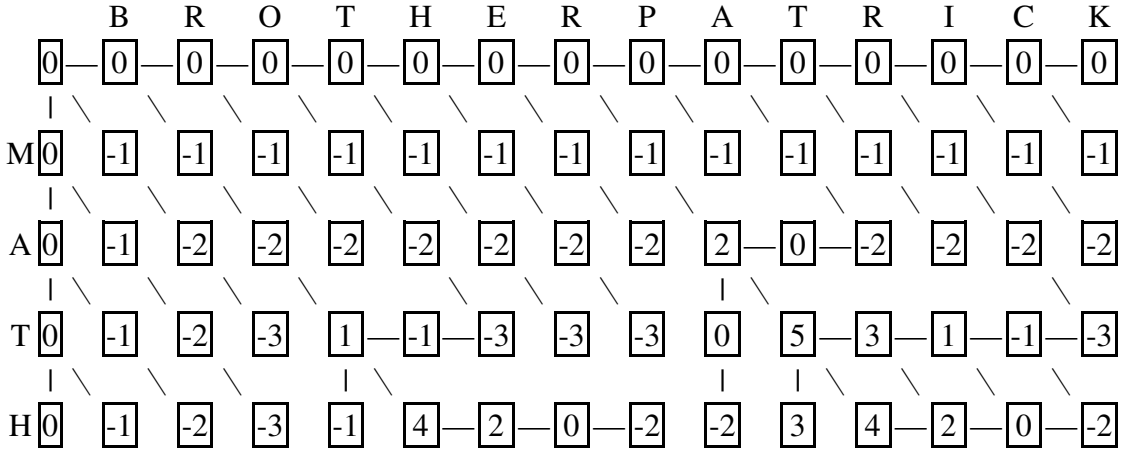


6. This is another scoring matrix for aligning the same pair of strings:
   a. Does it charge for initial gaps?
   b. How many points are given for a matched column?
   c. How many points are given for a mismatched column?
   d. Give all optimal alignments if there is no charge for initial gaps but there 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.

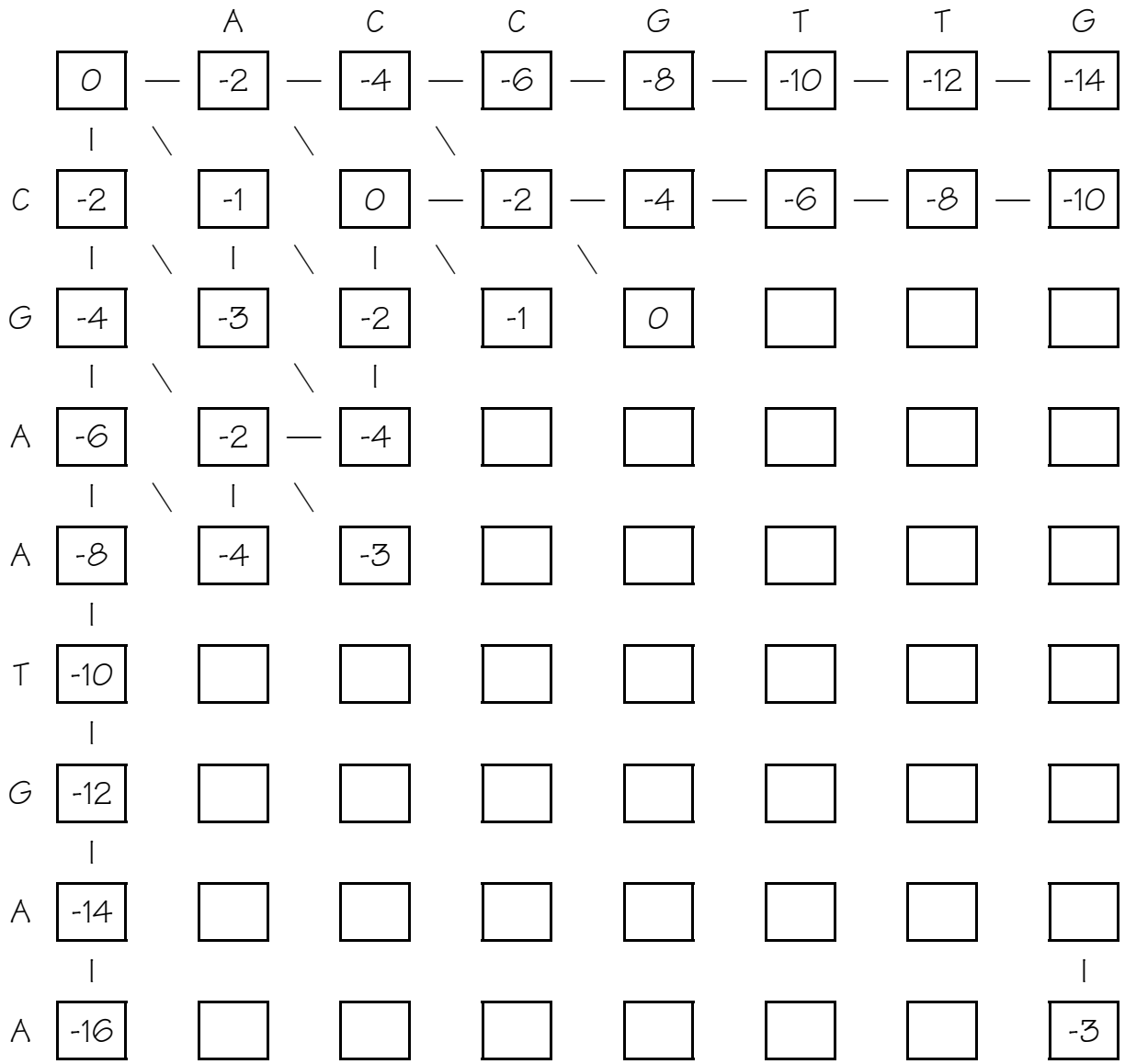


7. Suppose that in the previous problem we wished to align the same pair of strings using the same scoring system, except that gaps at the end of "BROTHERPATRICK" cost "-2" and gaps at the end of "MATH" cost "-1." How would the scoring matrix be altered? In particular, show that the resulting matrix yields a unique optimal alignment. Here are two copies of the matrix to play with:



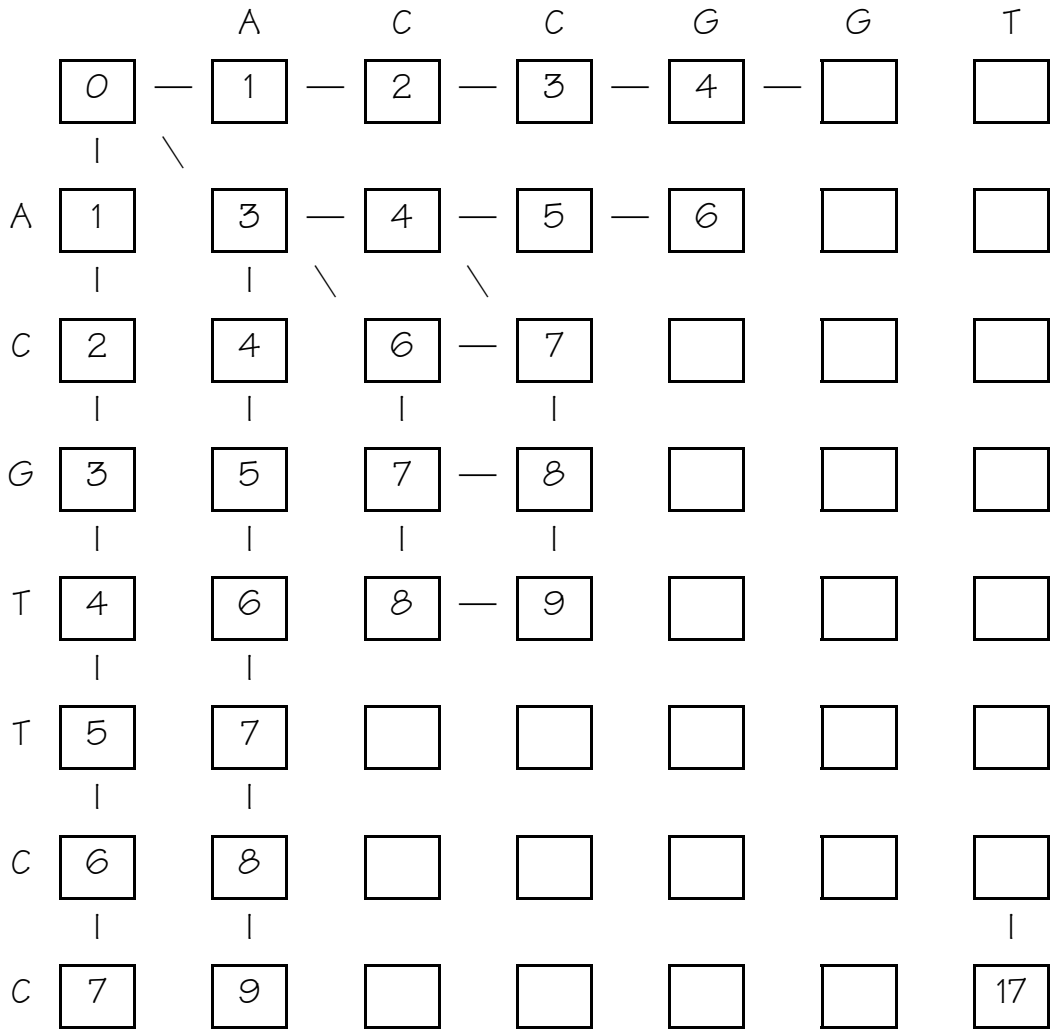


8. Here is the start of an alignment between "ACCGTTG" and "CGAATGAA" with match score 2, mismatch penalty -1 and gap penalty -2. Feel free to finish it.



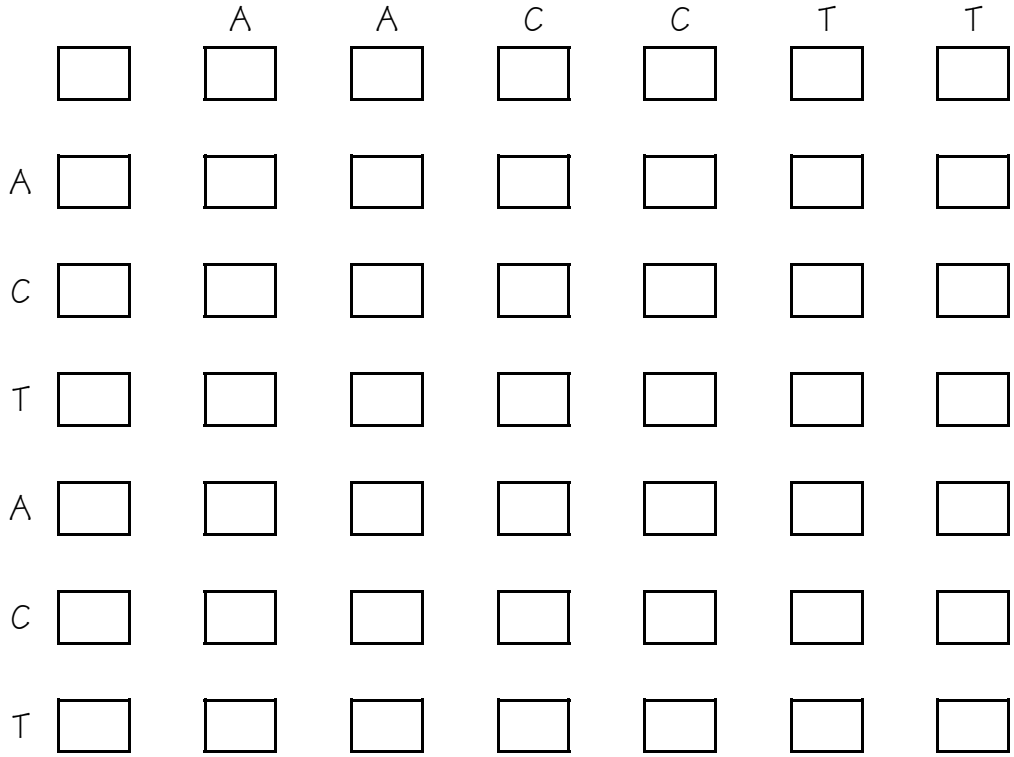
9. Give all optimal alignments between "ACCGTTG" and "CGAATGAA" with match score 2, mismatch penalty -1 and gap penalty -2.

10. Two DNA sequences derived from a common ancestor in an environment in which deletions were much more likely than point mutations. To reflect this in an alignment, a researcher assigns a match score of +3, a mismatch score of -1 and a gap "penalty" of +1. Here is the resulting scoring matrix. You might enjoy finishing it.



11. Prove that under the (very artificial) scoring system given in the previous problem, an optimal alignment of any two strings will never align two mismatched bases. In fact, what relationship between the gap penalty and the mismatch penalty will guarantee this behavior?

12. Here are two strings: "AACCTT" and "ACTACT"
    a. Align them using the following scoring system: match = +2, mismatch = -3, initial gaps = -1 and all other gaps = -2.
    b. How many optimal alignments are there?
    c. Show the optimal alignments



13. Under an alignment of two nucleotide strings with match score +1, mismatch penalty -1 and gap penalty -2, how large could the difference be between the values in the two boxes shaded below? Could they possibly be the same?