Objectives

- Dynamic Programming
  - Sequence Alignment

Review

- What is the key idea behind dynamic programming solutions?
- What is the process for solving the problems?

What was the Key to Solving each of these Problems?

- Weighted interval scheduling
- Segmented least squares
- Knapsack

What was the Key to Solving each of these Problems?

- Weighted interval scheduling
  - Binary decision: job was in or wasn't
  - Know conflicts→ reduce problem
- Segmented least squares
  - Knew last point was definitely in one segment
    - Could reduce
  - Multiway decision→ many possibilities for segment starting point
- Knapsack
  - If select an item, reduce available size by item's size
    - Find opt solution for smaller weight, remaining items

Analyzing Solution

```
Input: N, w1,…,wN, v1,…,vN

for w = 0 to W                                          O(W)
   M[0, w] = 0

for i = 1 to N      # for all items
   for w = 1 to W  # for all possible weights           O(N W)
      if wi > w :  # item's weight is more than available
         M[i, w] = M[i-1, w]
      else
         M[i, w] = max{ M[i-1, w], vi + M[i-1, w-wi] }

return M[n, W]
```

Why did we compute the best solution for all weights?

Knapsack Algorithm

W + 1 →

n + 1 ↓

	0	1	2	3	4	5	6	7	8	9	10	11
ϕ	0	0	0	0	0	0	0	0	0	0	0	0
{ 1 }	0	1	1	1	1	1	1	1	1	1	1	1
{ 1, 2 }	0	1	6	7	7	7	7	7	7	7	7	7
{ 1, 2, 3 }	0	1	6	7	7	18	19	24	25	25	25	25
{ 1, 2, 3, 4 }	0	1	6	7	7	18	22	24	28	29	29	40
{ 1, 2, 3, 4, 5 }	0	1	6	7	7	18	22	28	29	34	35	40

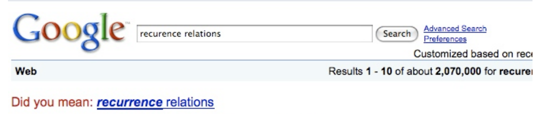
OPT: 40 = 22 + 18
Solution={4, 3}

W = 11

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

SEQUENCE ALIGNMENT

Has This Ever Happened To You?



How does Google know what I really meant?

String Similarity

- How similar are two strings?
  - ocurrance
  - occurrence
- We intuitively can tell that these two are similar
  - Ideas about a systematic measurement?

String Similarity

- How similar are two strings?
  - ocurrance
  - occurrence
- Measurements
  - Gap (-): add a letter
  - Mismatch

o c u r r a n c e -
o c c u r r e n c e
6 mismatches, 1 gap

o c - u r r a n c e
o c c u r r e n c e
1 mismatch, 1 gap

o c - u r r - a n c e
o c c u r r e - n c e
0 mismatches, 3 gaps

Which is the best alignment?

Edit Distance

- [Levenshtein 1966, Needleman-Wunsch 1970]
  - **Gap** penalty: δ
  - **Mismatch** penalty: α_{pq}
    - If p and q are the same, then mismatch penalty is 0
  - **Cost** = sum of gap and mismatch penalties

Parameters allow us to tweak cost

C T G A C C T A C C T
C C T G A C T A C A T
$\alpha_{TC} + \alpha_{GT} + \alpha_{AG} + 2\alpha_{CA}$

- C T G A C C T A C C T
C C T G A C - T A C A T
$2\delta + \alpha_{CA}$

Sequence Alignment

- **Goal**: Given two strings $X = x_1 x_2 \ldots x_m$ and $Y = y_1 y_2 \ldots y_n$, find alignment of minimum cost
- An *alignment* M is a set of ordered pairs x_i-y_j such that each item occurs in at most one pair and **no** crossings
- The pair x_i-y_j and $x_{i'}$-$y_{j'}$ *cross* if i < i', but j > j'.

o c c u r r e n c e
o c c u r r e n c e
crossing

o c c u r r e n c e
o c c u r r e n c e
2 mismatches

Sequence Alignment Example

- X = CTACCG
- Y = TACTG
- **Solution**: M = x_2-y_1, x_3-y_2, x_4-y_3, x_5-y_4, x_6-y_6

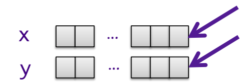
x_1	x_2	x_3	x_4	x_5		x_6
C	T	A	C	C	-	G
-	T	A	C	A	T	G
	y_1	y_2	y_3	y_4	y_5	y_6

$$\text{cost}(M) = \underbrace{\sum_{(x_i, y_j) \in M} \alpha_{x_i y_j}}_{\text{mismatch}} + \underbrace{\sum_{i : x_i \text{ unmatched}} \delta + \sum_{j : y_j \text{ unmatched}} \delta}_{\text{gap}}$$

Recall: mismatch penalty is 0 if x_i and y_j are the same

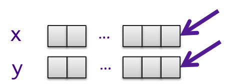
Sequence Alignment Case Analysis

- Consider the last character of the strings X and Y: x_M and y_N
  - M and N are not necessarily equal
- What are the possibilities for x_M and y_N in terms of the alignment?



Sequence Alignment Case Analysis

- Consider last character of strings X and Y: x_M and y_N
  - Case 1: x_M and y_N are aligned
  - Case 2: x_M is not matched
  - Case 3: y_N is not matched



Formulate the optimal solution's value

Sequence Alignment Case Analysis

- Consider last character of strings X and Y: x_M and y_N
  - Case 1: x_M and y_N are aligned
  - Case 2: x_M is not matched
  - Case 3: y_N is not matched

What are the costs for these cases?

- $\text{OPT}(\overset{x}{i}, \overset{y}{j})$ = min cost of aligning strings $x_1 x_2 \ldots x_i$ and $y_1 y_2 \ldots y_j$

Sequence Alignment Cost Analysis

- Consider last character of strings X and Y: x_M and y_N
  - Case 1: x_M and y_N are aligned
    - Pay mismatch for x_M-y_N + min cost of aligning rest of strings
    - $\text{OPT}(M, N) = \alpha_{x_m y_n} + \text{OPT}(M-1, N-1)$
  - Case 2: x_M is not matched
    - Pay gap for x_M + min cost of aligning rest of strings
    - $\text{OPT}(M, N) = \delta + \text{OPT}(M-1, N)$
  - Case 3: y_N is not matched
    - Pay gap for y_N + min cost of aligning rest of strings
    - $\text{OPT}(M, N) = \delta + \text{OPT}(M, N-1)$

What else do we need to consider?

Sequence Alignment Cost Analysis

- Base costs? → i or j is 0
  - What happens when we run out of letters in one string before the other?

```
X = CTACCG
Y = TACTG
```

Sequence Alignment: Problem Structure

Gaps for remainder of Y

$$OPT(i,j)=\begin{cases} j\delta & \text{if } i=0\\ \min\begin{cases}\alpha_{x_i y_j}+OPT(i-1,j-1)\\ \delta+OPT(i-1,j)\\ \delta+OPT(i,j-1)\end{cases} & \text{otherwise}\\ i\delta & \text{if } j=0\end{cases}$$

Ran out of 1st string

Ran out of 2nd string

Gaps for remainder of X

Sequence Alignment: Algorithm

Cost parameters

```
Sequence-Alignment(m, n, x_1x_2...x_m, y_1y_2...y_n, δ, α)
   for i = 0 to m
      M[i, 0] = iδ
   for j = 0 to n
      M[0, j] = jδ

   for i = 1 to m
      for j = 1 to n
         M[i, j] = min(α[x_i, y_j] + M[i-1, j-1],
                          δ + M[i-1, j],
                          δ + M[i, j-1])
   return M[m, n]
```


Sequence Alignment: Algorithm

Cost parameters

```
Sequence-Alignment(m, n, x_1x_2...x_m, y_1y_2...y_n, δ, α)
   for i = 0 to m
      M[i, 0] = iδ
   for j = 0 to n
      M[0, j] = jδ

   for i = 1 to m
      for j = 1 to n
         M[i, j] = min(α[x_i, y_j] + M[i-1, j-1],
                          δ + M[i-1, j],
                          δ + M[i, j-1])
   return M[m, n]
```

Ran out of y

Ran out of x

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
b					
o					
o					
t					

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2				
o	4				
o	6				
t	8				

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2	0	2	4	6
o	4				
o	6				
t	8				

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2	0	2	4	6
o	4	2	1	3	5
o	6				
t	8				

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2	0	2	4	6
o	4	2	1	3	5
o	6	4	3	2	4
t	8				

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2	0	2	4	6
o	4	2	1	3	5
o	6	4	3	2	4
t	8	6	5	4	2

Example

X = bait **Y = boot**

α = 1, for vowel mismatch
α = 2, for other mismatches
δ = 2

j

i

		b	a	i	t
	0	2	4	6	8
b	2	0	2	4	6
o	4	2	1	3	5
o	6	4	3	2	4
t	8	6	5	4	2

Sequence Alignment: Analysis

```
Sequence-Alignment(m, n, x1x2...xm, y1y2...yn, δ, α)
   for i = 0 to m
      M[i, 0] = iδ
   for j = 0 to n
      M[0, j] = jδ

   for i = 1 to m
      for j = 1 to n
         M[i, j] = min(α[xi, yj] + M[i-1, j-1],
                       δ + M[i-1, j],
                       δ + M[i, j-1])
   return M[m, n]
```

O(mn)

Costs?

Oracle of Bacon

The Oracle cannot find "*Dan Akroid.*" You have probably misspelled your entry, but it's also possible that the actor or actress you seek is not in our database. Below are 12 close matches. If you don't find what you're looking for in that list, try searching at www.imdb.com.

- Dan Aykroyd
- Dan Akre
- Dan Alford
- Dan Arkin
- Dan Broad
- Dan Krohn
- Dan Kroon
- Dan Marois
- Dan Sakraida
- Dan Stroud
- Dana Roi
- Daniel Krogh