CSE 417: Algorithms and Computational Complexity

W. L. Ruzzo
Dynamic Programming, II
RNA Folding
Outline

A few (well, ~25) slides on applications of dynamic programming in biology (You might enjoy a slightly deeper look at the use of some of the algorithms we study)

- Sequence alignment
- RNA structure

Algorithms for RNA structure (probable HW)
Application 1: Sequence Search
Moore’s Law
Growth of GenBank (Base Pairs)

Sequencing Costs Outpace Moore’s Law
A Database Search

go to, e.g., [http://www.uniprot.org/](http://www.uniprot.org/), “blast” tab, and paste in this:

>sp|P15172|MYOD1_HUMAN Myoblast determination protein 1 OS=Homo sapiens GN=MYOD1 PE=1 SV=3
MELLSPPLRDVDLTAPDGSLCSFATTDDFYDDPCFDSPDLRFFEDLDPRLMHVGA LKKPE
EHSHPAAVHPAPGAREDEHEVRAQPSGHQQAGRLWACKACKRKRTTNADRRKAATMRERR
RLSKVNEAFETLKRCTSSNPQRLPKVEILRNNAIYIEGLQALLRDPDAAPPGAAAAYFA
PGPLPPGRGGEHYSGDSDASSPRSCSDDGMDYSGPPSGARRNNCYEGAYYNEAPSEPRP
GKSAAVSSLDCCLSSIVERISTESPAAAPALLLADVSESPPPPQEAAPSEGESSGDPTQS
PDAAPQCPAGANPNPIYQVL
A Few seconds Later…

...And 1000’s more…

<table>
<thead>
<tr>
<th>Accession</th>
<th>Entry name</th>
<th>0Query hit320</th>
<th>0Match hit (sqrt scale)17392</th>
<th>Name (Organism)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P15172</td>
<td>MYOD1_HUMAN</td>
<td>human</td>
<td></td>
<td>Myoblast determination protein 1 (Homo sapiens)</td>
</tr>
<tr>
<td>B2RC72</td>
<td>B2RC72_HUMAN</td>
<td>human</td>
<td></td>
<td>cDNA, FLJ95884, highly similar to Hom... (Homo sapiens)</td>
</tr>
<tr>
<td>E2RT59</td>
<td>E2RT59_CANFA</td>
<td>dog</td>
<td></td>
<td>Uncharacterized protein (Canis familiaris)</td>
</tr>
<tr>
<td>P49811</td>
<td>MYOD1_PIG</td>
<td>pig</td>
<td></td>
<td>Myoblast determination protein 1 (Sus scrofa)</td>
</tr>
<tr>
<td>D2KP19</td>
<td>D2KP19_PIG</td>
<td>pig</td>
<td></td>
<td>Myogenic differentiation 1 (Sus scrofa)</td>
</tr>
<tr>
<td>F1S9A9</td>
<td>F1S9A9_PIG</td>
<td>pig</td>
<td></td>
<td>Uncharacterized protein (Sus scrofa)</td>
</tr>
<tr>
<td>D2I0V4</td>
<td>D2I0V4_AILME</td>
<td>panda</td>
<td></td>
<td>Putative uncharacterized protein (Ailuropoda melanoleuca)</td>
</tr>
<tr>
<td>P29331</td>
<td>MYOD1_SHEEP</td>
<td>sheep</td>
<td></td>
<td>Myoblast determination protein 1 (Ovis aries)</td>
</tr>
<tr>
<td>D2SP11</td>
<td>D2SP11_BUBBU</td>
<td>water buffalo</td>
<td></td>
<td>Myogenic factor MYOD1 (Bubalus bubalis)</td>
</tr>
<tr>
<td>Q0VBX9</td>
<td>Q0VBX9_BOVIN</td>
<td>cow</td>
<td></td>
<td>Myogenic differentiation 1 (Bos taurus)</td>
</tr>
<tr>
<td>Q7YS82</td>
<td>MYOD1_BOVIN</td>
<td>cow</td>
<td></td>
<td>Myoblast determination protein 1 (Bos taurus)</td>
</tr>
<tr>
<td>Q8C6B1</td>
<td>Q8C6B1_MOUSE</td>
<td>mouse</td>
<td></td>
<td>Myogenic differentiation 1 (Mus musculus)</td>
</tr>
<tr>
<td>A0JP9K</td>
<td>A0JP9K_RAT</td>
<td>rat</td>
<td></td>
<td>Myoblast determination protein 1 (Rattus norvegicus)</td>
</tr>
<tr>
<td>Q02346</td>
<td>MYOD1_RAT</td>
<td>rat</td>
<td></td>
<td>Myoblast determination protein 1 (Rattus norvegicus)</td>
</tr>
<tr>
<td>P10085</td>
<td>MYOD1_MOUSE</td>
<td>mouse</td>
<td></td>
<td>Myoblast determination protein 1 (Mus musculus)</td>
</tr>
<tr>
<td>Q6DTY5</td>
<td>Q6DTY5_PIG</td>
<td>pig</td>
<td></td>
<td>Eukaryotic myogenic factor MYF-3 (Sus scrofa)</td>
</tr>
<tr>
<td>P21572</td>
<td>MYOD1_COTJA</td>
<td>quail</td>
<td></td>
<td>Myoblast determination protein 1 homolog (Coturnix coturnix japonica)</td>
</tr>
<tr>
<td>Q6DV59</td>
<td>Q6DV59_MELGA</td>
<td>turkey</td>
<td></td>
<td>MyoD (Meleagris gallopavo)</td>
</tr>
<tr>
<td>P16075</td>
<td>MYOD1_CHICK</td>
<td>chicken</td>
<td></td>
<td>Myoblast determination protein 1 homolog (Gallus gallus)</td>
</tr>
<tr>
<td>C5J072</td>
<td>C5J072_CHICK</td>
<td>chicken</td>
<td></td>
<td>Myogenic differentiation 1 (Gallus gallus)</td>
</tr>
<tr>
<td>C3U011</td>
<td>C3U011_ANAPL</td>
<td>duck</td>
<td></td>
<td>Myogenic differentiation 1 (Anas platyrhynchos)</td>
</tr>
<tr>
<td>F1N0M3</td>
<td>F1N0M3_CHICK</td>
<td>chicken</td>
<td></td>
<td>Uncharacterized protein (Gallus gallus)</td>
</tr>
<tr>
<td>F1N0X5</td>
<td>F1N0X5_CHICK</td>
<td>chicken</td>
<td></td>
<td>Uncharacterized protein (Gallus gallus)</td>
</tr>
<tr>
<td>P13904</td>
<td>MYODA_XENLA</td>
<td>frog</td>
<td></td>
<td>Myoblast determination protein 1 homolog A (Xenopus laevis)</td>
</tr>
<tr>
<td>Q8AVZ0</td>
<td>Q8AVZ0_XENLA</td>
<td>frog</td>
<td></td>
<td>Myod1-a protein (Xenopus laevis)</td>
</tr>
<tr>
<td>Q7T109</td>
<td>Q7T109_XENTR</td>
<td>frog</td>
<td></td>
<td>MyoD protein (Xenopus tropicalis)</td>
</tr>
<tr>
<td>Accession</td>
<td>Entry name</td>
<td>Status</td>
<td>Protein names</td>
<td>Organism</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>--------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>Q7T109</td>
<td>Q7T109_XENTR</td>
<td>⭐️</td>
<td>MyoD protein</td>
<td>Xenopus tropicalis (Western clawed frog) (Xenopus tropicalis)</td>
</tr>
</tbody>
</table>

Alignment 1 against Q7T109

<table>
<thead>
<tr>
<th>Score</th>
<th>E-value</th>
<th>Identity</th>
<th>Positives</th>
<th>Match length</th>
</tr>
</thead>
<tbody>
<tr>
<td>964</td>
<td>1.0 x 10^-102</td>
<td>64.0%</td>
<td>74.0%</td>
<td>288</td>
</tr>
</tbody>
</table>

Position
Q7T109 matches from 1 to 288 (288AA), in the query sequence from 1 to 320 (320AA)

Graphical
1. MELLSPPLRDVDLTAPIPDGSLSFATTDFFYDDPCFDSPDLRFDFEDLDPRLHMVGALLKPE
   MELL PPLRD+++T +GLCSF T DDFYDDPCF++ D+ FFEDLDPRL+HV ALLKPE
   60  P15172

1. MELLPPPLRDMEVT---EGSLCSFPTPDFFYDDPCFTSDMSFEDLDPRLVHV---ALLKPE
   57  Q7T109

61. EHSHPFPAVHPAPGPAGREDHEVRAPSGHHAQGRCLLWACKACKRKTTNADRRKAA
   HEDHEVRAPSGHHAQGRCLLWACKACKRKTTNADDRAA
   120 P15172

58. DPHH---------NEDEHVRAPSGHHAQGRCLLWACKACKRKTTNADRRKAA
   106 Q7T109

121. RLSKVNEAFETLKRCTSSNPQLPKVEILRNAIRYEGLQALLRDQDAAPPGAAAAYFA
   RLSKVNEAFETLKRCTS+NPNQLPKVEILRNAIRE LQ+LLR Q+ +FY
   180 P15172

107. RLSKVNEAFETLKRCTSNPQLPKVEILRNAIRELSQSLLRQGe-------ESFY-
   158 Q7T109

181. PGPLPPGRGGEHYSGDSDASSPNSCSDGMMDYSGPSGARRRNCYEGAYYNEAPSEPRP
   P+ EHYSGDSDASSPNSCSDGM DYS PP G+RRRN Y++Y+++P+ R
   240 P15172

159. --PVL------EHYSGDSDASSPNSCSDGMTDYS--PPCGSSRRNSYDSSFYS DSPNLRL
   210 Q7T109

241. GKSAAVSSLDCNSSIVESTESPAAAPALLLADVSESPPRSEQAAPSEGES---SGDP
   GKS+ +SSLDCNSSIVESTESP P + AD SE P +P +GE+ SG
   297 P15172

211. GKSVSSLDCNSSIVESTESPVCVPVPAADSGSEGSP-----CSPLQGETLSESIGII
   265 Q7T109
The foregoing search capability is a *huge* deal

the “google” of molecular biology

millions of searches daily

biologists (not just “computational” biologists) use this routinely

it connects information about *all* living things

(dynamic programming)

Time permitting, more on algorithm later …
Application 2: RNA Structure
As shown, the two strands coil about each other in a fashion such that all the bases project inward toward the helix axis. The two strands are held together by hydrogen bonds (pink rods) linking each base projecting from one backbone to its so-called complementary base projecting from the other backbone. The base A always bonds to T (A and T are comple-
Fig. 2. The arrows show the situation as it seemed in 1958. Solid arrows represent probable transfers, dotted arrows possible transfers. The absent arrows (compare Fig. 1) represent the impossible transfers postulated by the central dogma. They are the three possible arrows starting from protein.
Non-coding RNA

Messenger RNA - codes for proteins
Non-coding RNA - all the rest

Before, say, mid 1990’s, 1-2 dozen known
(critically important, but narrow roles: e.g., tRNA)

Since mid 90’s dramatic discoveries

Regulation, transport, stability/degradation
E.g. “miRNA”: >1000 in humans; regulate >50% of genes
E.g. “riboswitches”: 10000’s in bacteria

By some estimates, ncRNA >> mRNA
DNA structure: dull

5’…ACCGCTAGATG…3’

3’…TGGCGATCTAC…5’
RNA Secondary Structure: RNA makes helices too

Base pairs

A-U
C-G

Usually single stranded
RNA Secondary Structure:

Not everything, but important, easier than 3d
Why is structure important?

• For protein-coding, similarity in sequence is a powerful tool for finding related sequences
  – e.g. “hemoglobin,” “MyoD” and many others are easily recognized in all animals

• For many non-coding RNAs, different sequences can have the same structure, and structure is most important for function.
  – So, using structure plus sequence, can find related sequences at much greater evolutionary distances
  – 2 Examples below
6S mimics an open promoter

Barrick et al. RNA 2005
Trotochaud et al. NSMB 2005
Willkomm et al. NAR 2005
“Riboswitches”: small molecule sensors & gene on/off switches
Origin of Life?

Life needs

- information carrier: DNA
- molecular machines, like enzymes: Protein
- making proteins needs DNA + RNA + proteins
- making (duplicating) DNA needs proteins

Horrible circularities! How could it have arisen in an abiotic environment?
Origin of Life?

RNA can carry information, too
   RNA double helix; RNA-directed RNA polymerase
RNA can form complex structures
RNA enzymes exist (ribozymes)
RNA can control, do logic (riboswitches)

The “RNA world” hypothesis:
1st life was RNA-based
6.5 RNA Secondary Structure

Nussinov’s Algorithm – core technology for RNA structure prediction
RNA Secondary Structure

RNA. String $B = b_1 b_2 \ldots b_n$ over alphabet \{A, C, G, U\}.

Secondary structure. RNA is usually single-stranded, and tends to loop back and form base pairs with itself. This structure is essential for understanding molecular behavior.

Ex: GUCGAUUGAGCGAAUGUAACAACGUGGCUACGGCGAGA
RNA Secondary Structure (≈ oversimplified)

RNA: String \( B = b_1b_2\ldots b_n \) over alphabet \( \{ \text{A, C, G, U} \} \).

Secondary structure: A set of pairs \( S = \{ (b_i, b_j) \} \) satisfying:

- [Watson-Crick Pairing.]
  - \( S \) is a matching: each base pairs with \( \leq \) other, and
  - each pair in \( S \) is a Watson-Crick pair: A-U, U-A, C-G, or G-C.

- [No sharp turns.] Pairs are separated by \( \geq 4 \) intervening bases.
  - If \( (b_i, b_j) \in S \), then \( i < j - 4 \).

- [Non-crossing.] If \( (b_i, b_j) \) and \( (b_k, b_l) \) are two pairs in \( S \), then we cannot have \( i < k < j < l \). (Violation is called a pseudoknot.)

What’s Best: RNA will form the structure that minimizes free energy.

Goal: find a secondary structure \( S \) maximizing the number of base pairs.

T in DNA → U in RNA

approximated by maximizing number of base pairs
RNA Secondary Structure: Examples

Examples.

- **Base pair**
  - ![Base pair example](image)
  - **OK**

- **Sharp turn**
  - ![Sharp turn example](image)
  - **OK**

- **Crossings**
  - ![Crossings example](image)
  - **Not allowed**
RNA Secondary Structure: Subproblems

First attempt. \( \text{OPT}[j] = \) maximum number of base pairs in a secondary structure of the substring \( b_1b_2 \ldots b_j \).

\[
\text{match } b_t \text{ and } b_j
\]

Results in two sub-problems.
- Find secondary structure in: \( b_1b_2 \ldots b_{t-1} \). \( \leftarrow \text{OPT}(t-1) \); good!
- Find secondary structure in: \( b_{t+1}b_{t+2} \ldots b_{j-1} \). \( \leftarrow \) DIFFICULTY: this isn’t “OPT” of anything; need more flexible set of sub-problems
Dynamic Programming Over Intervals: (R. Nussinov’s algorithm)

Notation. \( \text{OPT}[i, j] = \) maximum number of base pairs in a secondary structure of the substring \( b_i b_{i+1} \ldots b_j \).

- Case 1a. If \( i \geq j - 4 \) (and base \( b_j \) is not paired):
  \[ \text{OPT}[i, j] = 0 \text{ by no-sharp turns condition.} \]

- Case 1b. If \( i < j - 4 \), but base \( b_j \) is not paired:
  \[ \text{OPT}[i, j] = \text{OPT}[i, j-1] \]

- Case 2. Base \( b_j \) pairs with \( b_t \) for some \( i \leq t < j - 4 \). Non-crossing constraint decouples resulting sub-problems
  \[ \text{OPT}[i, j] = 1 + \max_t \{ \text{OPT}[i, t-1] + \text{OPT}[t+1, j-1] \} \]

Key point: Either last base is unpaired (case 1a,b) or paired (case 2)

take max over \( t \) such that \( i \leq t < j - 4 \) and \( b_t \) and \( b_j \) are Watson-Crick complements

omit when \( t == i \) (see next slide)
“Optimal pairing of \(b_i \ldots b_j\)"

Two possibilities:

j Unpaired:
Find best pairing of \(b_i \ldots b_{j-1}\)

j Paired (with some \(t\)):
Find best \(b_i \ldots b_{t-1}\) +
best \(b_{t+1} \ldots b_{j-1}\) plus 1

Why is it slow?
Why do pseudoknots matter?
Bottom Up Dynamic Programming Over Intervals

Q. What order to solve the sub-problems?
A1. Book way—do shortest intervals first, then earliest start:

\[
\text{RNA}(b_1, \ldots, b_n) \{ \\
\text{for } k = 5, 6, \ldots, n-1 \\
\text{for } i = 1, 2, \ldots, n-k \\
\quad j = i + k \\
\quad \text{Compute } OPT[i, j] \\
\text{return } OPT[1, n]
\}
\]

A2. Slides way: earliest start first, then shortest intervals (next slides)

Running time. \(O(n^3)\) (either way)
Nussinov: Max Pairing

\[ \text{Opt}[i,j] = \# \text{ pairs in optimal pairing of } b_i \ldots b_j \]
\[ \text{Opt}[i,j] = 0 \text{ for all } i, j \text{ with } i \geq j-4; \text{ otherwise} \]
\[ \text{Opt}[i,j] = \max \text{ of:} \]
\[
\begin{cases}
\text{Opt}[i,j-1] \\
\max \left\{ \text{Opt}[i,t-1]+1+\text{Opt}[t+1,j-1] \middle| i \leq t < j-4 \text{ and } b_t-b_j \text{ may pair} \right\}
\end{cases}
\]
Another Computation Order

\[ \text{Opt}[i, j] = \text{optimal \# pairs in } b_i \ldots b_j \]

for \( j = 1 \) to \( n \)

for \( i = j \) downto \( 1 \)

\[ \text{Opt}[i, j] = 0 \text{ if } i \geq j-4 \text{ else:} \]

\[ \max \left\{ \text{Opt}[i, j-1] \right\} \]

\[ \max \left\{ \text{Opt}[i, t-1] + 1 + \text{Opt}[t+1, j-1] \mid \right. \]

\[ i \leq t < j-4 \text{ and } b_t - b_j \text{ may pair} \}

Time: \( O(n^3) \)
Which Pairs?

Usual dynamic programming “trace-back” tells you *which* base pairs are in the optimal solution, not just how many.

Details? : homework
Computing one cell: \( \text{OPT}[2, 18] = ? \)

\[
\begin{array}{cccccccccccccccccc}
( & ( & ( & \ldots & ) & ) & ) & ( & ( & ( & \ldots & ) & ) & ) & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 5 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 5 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 5 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 4 & 4 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 \\
\end{array}
\]

\( n = 20 \)

Case 1:
2 \( \geq \) 18-4? no.

Case 2:
B\(_{18}\) unpaired?
Always a possibility;
then \( \text{OPT}[2, 18] \geq 3 \)

\[
\text{GGAAAAACCCAAAGGGGU} \\
( (\ldots) ) (\ldots) \ldots
\]

\[
\text{OPT}(i, j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{l}
\text{OPT}[i, j - 1] \\
1 + \max_t (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1])
\end{array} \right\} & \text{otherwise}
\end{cases}
\]

(Examples here and below assume 1-based indexing)
Computing one cell: $\text{OPT}[2, 18] = ?$

$\text{OPT}(i, j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{l}
\text{OPT}[i, j - 1] \\
1 + \max_t (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1])
\end{array} \right\} & \text{otherwise}
\end{cases}$

Case 3, $2 \leq t < 18 - 4$: $t = 2$: no pair

$n = 20$
Computing one cell: $\text{OPT}[2,18] = ?$

\[
\text{OPT}(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{l}
\text{OPT}[i,j-1] \\
1 + \max_t (\text{OPT}[i,t-1] + \text{OPT}[t+1,j-1])
\end{array} \right. & \text{otherwise}
\end{cases}
\]

Case 3, $2 \leq t < 18 - 4$: $t = 3$: no pair
Computing one cell: $\text{OPT}[2, 18] = ?$

$$\text{OPT}(i, j) = \begin{cases} 0 & \text{if } i \geq j - 4 \\ \max \left\{ \text{OPT}[i, j - 1], 1 + \max_t (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1]) \right\} & \text{otherwise} \end{cases}$$

Case 3, $2 \leq t < 18 - 4$:

$t = 4$: yes pair

$\text{OPT}[2, 18] \geq 1 + 0 + 3$

$_{GGA\text{AAAACC}C\text{AAAGGGGU} \ldots (\ldots (((\ldots ))) )}$
Computing one cell: $\text{OPT}[2, 18] = ?$

$\text{OPT}(i, j) = \begin{cases} 
0 
& \text{if } i \geq j - 4 \\
\max \left\{ \text{OPT}[i, j - 1], \ 1 + \max_{t} (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1]) \right\} 
& \text{otherwise}
\end{cases}$
Computing one cell: $\text{OPT}[2, 18] = \text{?}$

Case 3, $2 \leq t < 18-4$

$t = 6$: yes pair

$\text{OPT}[2, 18] \geq 1 + 0 + 3$

$\text{GGAA} \text{AACCCAAAGGGGU}$

$\ldots \text{. . . . } ( ( ( ( . . . . ) ) ) )$

$\text{OPT}(i, j) = \begin{cases} 0 & \text{if } i \geq j - 4 \\ \max \left\{ \begin{array}{l} \text{OPT}[i, j - 1] \\ 1 + \max_t (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1]) \end{array} \right. & \text{otherwise} \end{cases}$
Computing one cell: $\text{OPT}[2, 18] = ?$

Case 3, $2 \leq t < 18 - 4$:
- $t = 7$: yes pair
- $\text{OPT}[2, 18] \geq 1 + 0 + 3$

$\text{OPT}(i, j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{l}
\text{OPT}[i, j - 1] \\
1 + \max_{t} (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1])
\end{array} \right. & \text{otherwise}
\end{cases}$
Computing one cell: \( \text{OPT}[2,18] = ? \)

\[
\text{OPT}(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{c}
\text{OPT}[i,j-1] \\
1 + \max_t (\text{OPT}[i,t-1] + \text{OPT}[t+1,j-1])
\end{array} \right\} & \text{otherwise}
\end{cases}
\]

Case 3, \( 2 \leq t < 18-4 \): 
\( t = 8 \): no pair
Computing one cell: OPT[2,18] = ?

**Case 3, 2 ≤ t < 18-4:**

**t = 11:** yes pair

OPT[2,18] ≥ 1 + 2 + 0

GGAAAAACCC AAAGGGGU
((......))  (......)

(not shown: t = 9, 10, 12, 13)

OPT(i, j) = \[
\begin{cases}
0 & \text{if } i \geq j - 4 \\
\max \left\{ \begin{array}{l}
\text{OPT}[i, j - 1] \\
1 + \max_t (\text{OPT}[i, t - 1] + \text{OPT}[t + 1, j - 1])
\end{array} \right. & \text{otherwise}
\end{cases}
\]
Computing one cell:
$OPT[2,18] = 4$

Overall, $Max = 4$
several ways, e.g.:
GGAAACCCAAAGGGGU

..(...(((....))))

tree shows trace back:
square = case 3
octagon = case 1

$OPT(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \{ \max_t (OPT[i,t-1] + OPT[t+1,j-1]) \} & \text{otherwise}
\end{cases}$
All 5 optimal structures on the above example

```
GGGAAAAACCCAAAGGGGUUU
...(((.(((.....))))))
...((.((((....))))))
...(.(((((....))])))
....((((((....))))))
(((....)))(((....)))
```

```
0 0 0 0 0 0 0 1 2 3 3 3 3 3 3 3 3 4 5 6
-7 0 0 0 0 0 0 1 2 2 2 2 2 3 3 3 4 5 6
-7-7 0 0 0 0 0 1 1 1 1 1 1 2 2 3 3 4 5 6
-7-7-7 0 0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6
-7-7-7-7 0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6
-7-7-7-7-7 0 0 0 0 0 0 0 0 1 2 2 3 4 5 5
-7-7-7-7-7-7 0 0 0 0 0 0 0 1 2 2 3 4 4 4
-7-7-7-7-7-7-7 0 0 0 0 0 0 1 2 2 3 3 3 3
-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 2 2 2 2 3
-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 1 1 2 3
-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 2 3
-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 2 2
-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 1
-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
n= 20 Pairs= 6 AltStructs= 5 0.000117 (sec. total)
Another Example

(Examples here and below assume 1-based indexing)
Another Trace Back Example

OPT(1,16) = 3:
CUCCGGUUUGCAUGUC
((. (....).).)..

\[
\text{OPT}(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left\{ \text{OPT}[i,j-1], 1 + \max_t (\text{OPT}[i,t-1] + \text{OPT}[t+1,j-1]) \right\} & \text{otherwise}
\end{cases}
\]
**A** L19 (rplS) mRNA leader

**B** Excess L19 represses L19 (RF00556; 555-559 similar)

**Example: Ribosomal Autoregulation**

**C** B. subtilis L19 mRNA leader

- **nucleotide identity**
  - **N** 97%
  - **N** 90%
  - **N** 75%

- **stem loop always present**
  - compatible mutations
  - compensatory mutations

- **G - C** Watson-Crick base pair
- **G • A** other base interaction
Covariation is strong evidence for base pairing.
Summary

RNA has important roles
  Beyond mRNA; many unexpected recent discoveries
Structure is critical to function
  True of other molecules, too
RNA secondary structure prediction is a key tool
Dynamic programming—useful accuracy, $O(n^3)$ time:
  Binary choice again: last base is paired or not
  Optimal substructure again: given last pair, optimally fold inside & outside separately
  Tabulate again: best folding of all substrings.