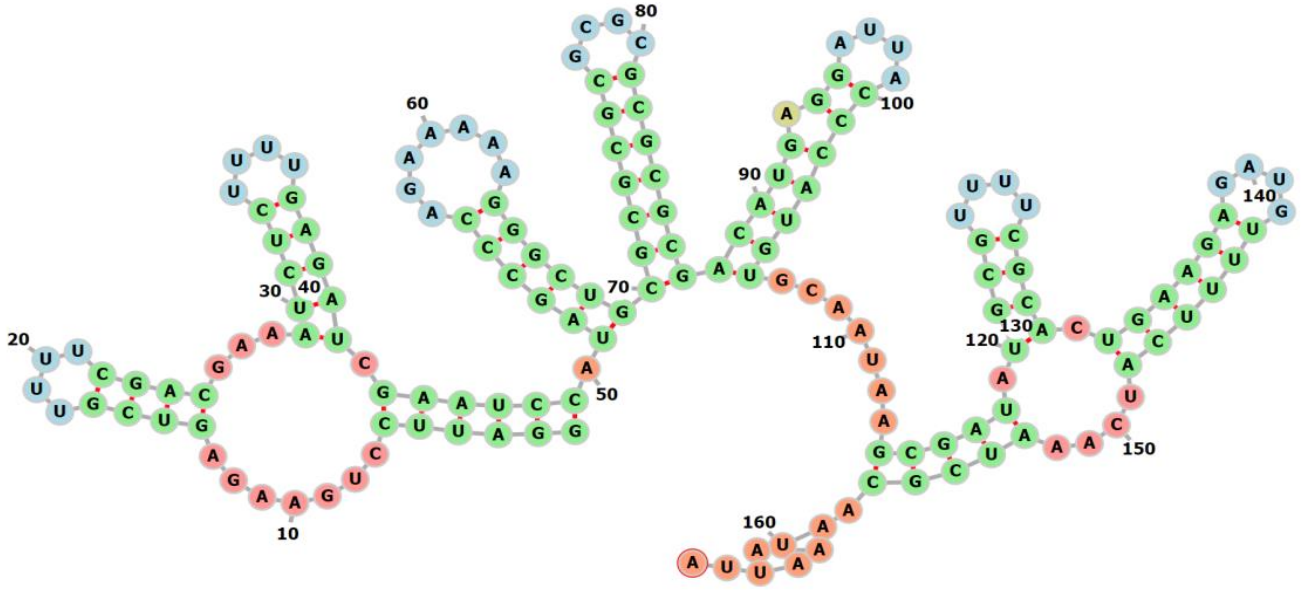
6.5 RNA Secondary Structure

20 aprile 2023

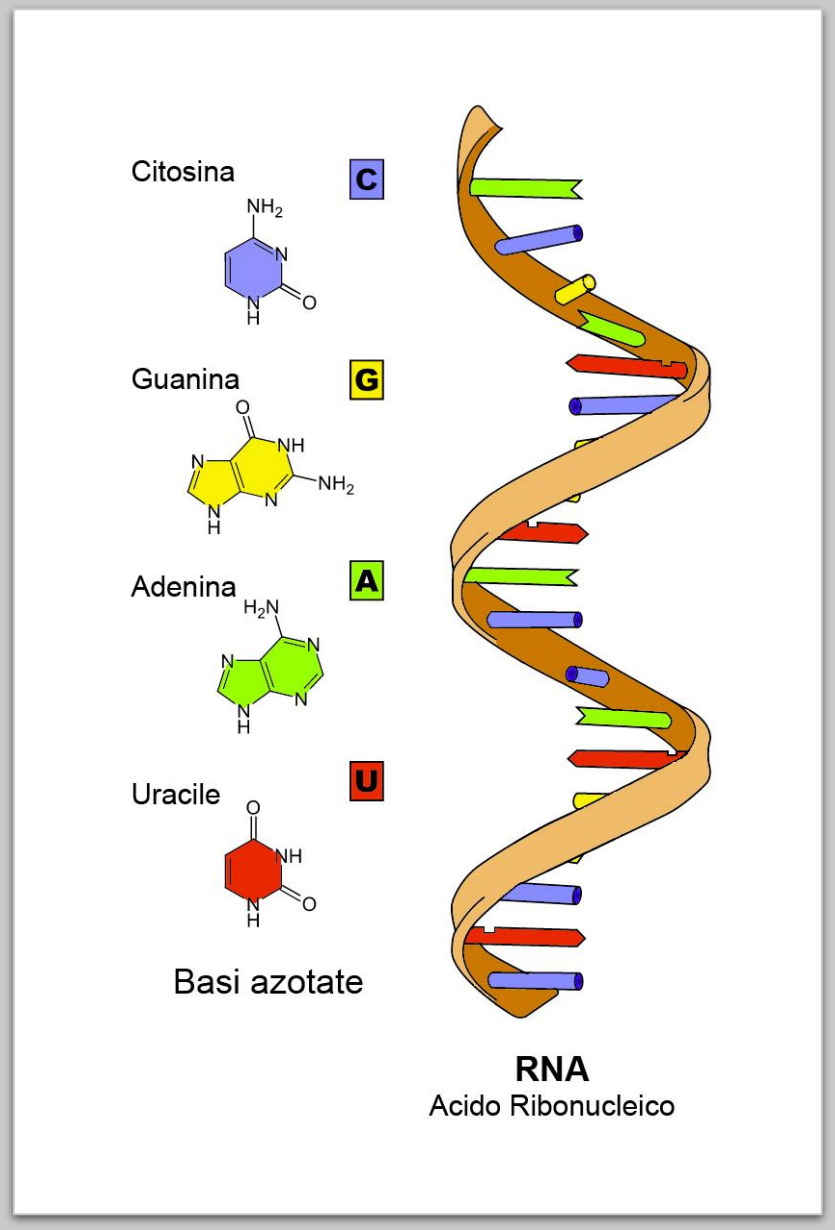


Struttura di una biomolecola

Biomolecola: DNA, RNA

Struttura primaria: descrizione esatta della sua composizione atomica e dei legami presenti fra gli atomi

Struttura secondaria: capacità di assumere una struttura spaziale regolare e ripetitiva



DNA

Struttura secondaria: doppia elica (Watson e Crick).

Ogni catena è composta da nucleotidi: A, C, G, T

A (adenina), C (citosina), G (guanina), T (timina)

Le catene sono connesse da basi complementari: A-T, C-G

Ribonucleic acid (RNA)

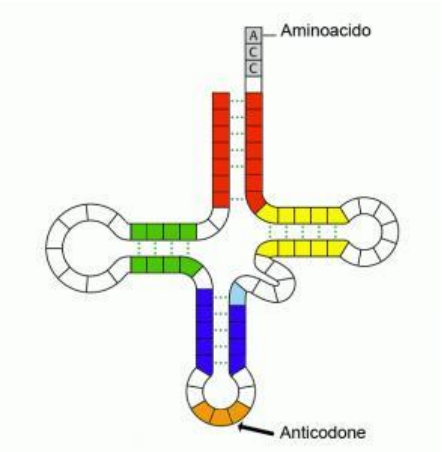
Simile al DNA.

Singola catena con 4 nucleotidi: adenine (A), cytosine (C), guanine (G), uracil (U).

RNA. Stringa $B = b_1 b_2 \ldots b_n$ su alfabeto { A, C, G, U }.

Struttura secondaria. RNA è una singola catena e tende a formare coppie di basi con se stessa. Questa struttura è essenziale per capire il comportamento delle molecole.

coppie di base complementari: A-U, C-G



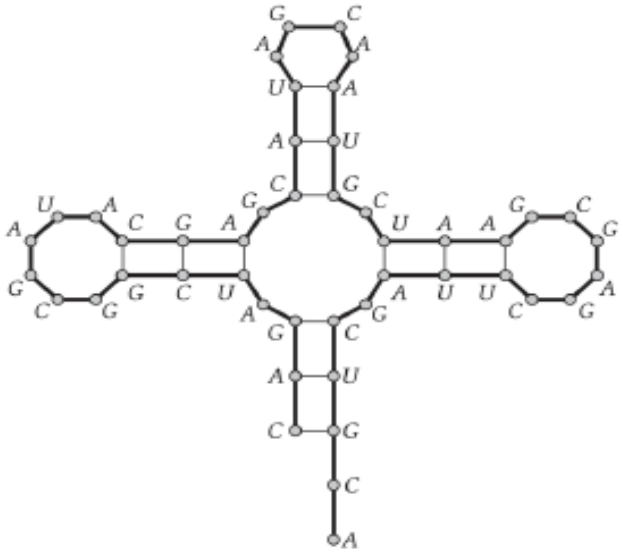


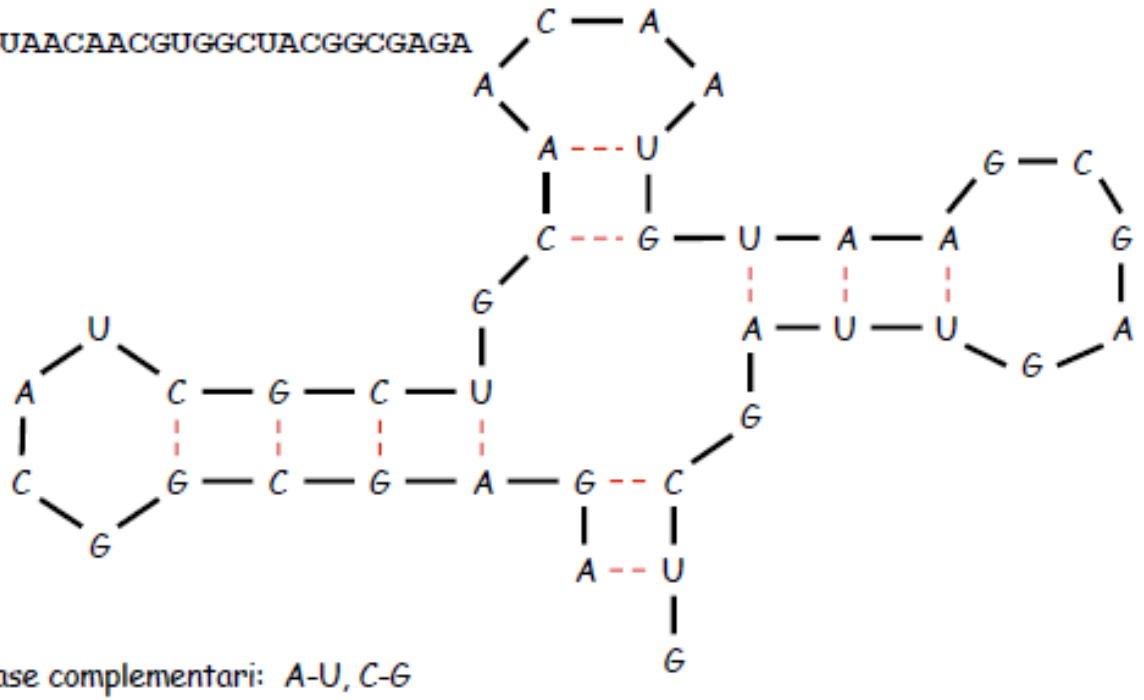
Figure 6.13 An RNA secondary structure. Thick lines connect adjacent elements of the sequence; thin lines indicate pairs of elements that are matched.

RNA Secondary Structure

RNA. Stringa B = $b_1 b_2 \ldots b_n$ su alfabeto { A, C, G, U }.

Struttura secondaria. RNA è una singola catena e tende a formare coppie di basi con se stessa. Questa struttura è essenziale per capire il comportamento delle molecole.

Esempio: GUCGAUUGAGCGAAUGUAACAACGUGGCUACGGCGAGA



coppie di base complementari: A-U, C-G

Per una stessa stringa di RNA possono esistere più strutture secondarie

Two views of RNA secondary structure

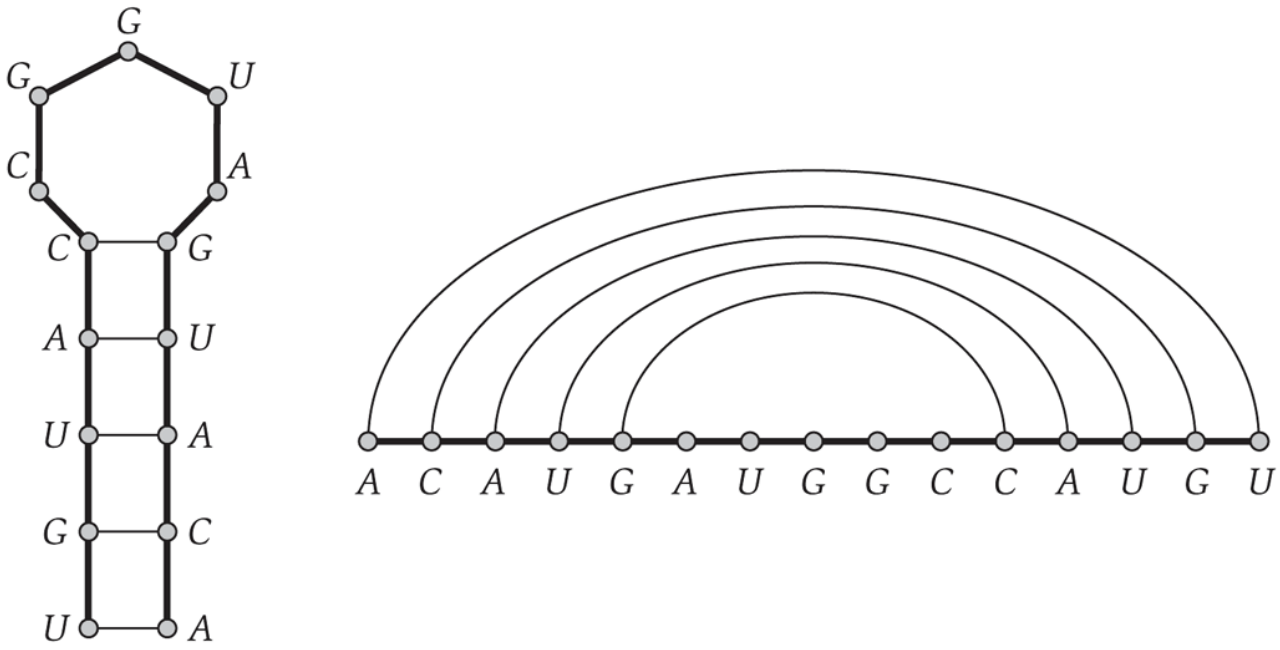
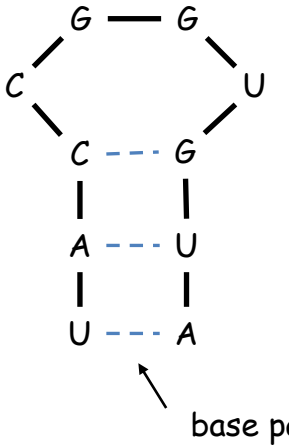
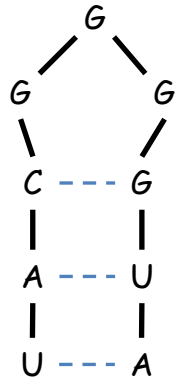


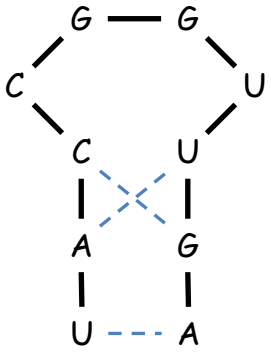
Figure 6.14 Two views of an RNA secondary structure. In the second view, (b), the string has been "stretched" lengthwise, and edges connecting matched pairs appear as noncrossing "bubbles" over the string.

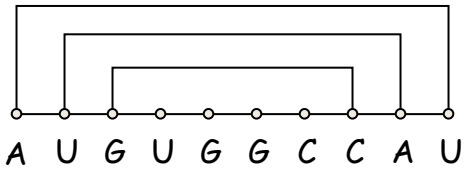
RNA Secondary Structure: Examples

Examples.

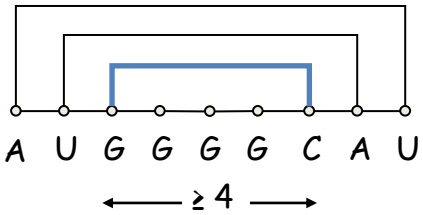




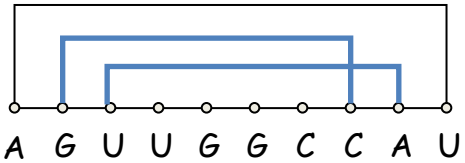




ok

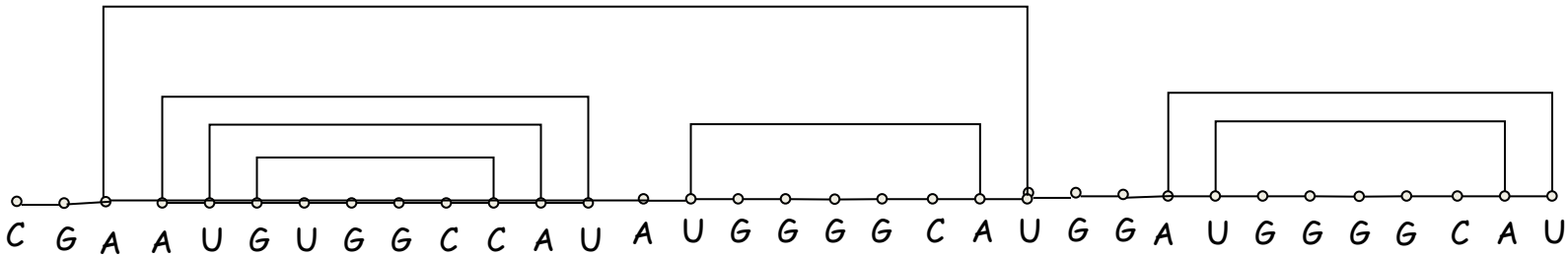


sharp turn: no!



crossing: no!

Un generico esempio



RNA Secondary Structure

Secondary structure. A set of pairs $S = \{ (b_i, b_j) \}$ that satisfy:

[Matching] no base appears in more than one pair.

[Watson-Crick.] S is a matching and each pair in S is a Watson-Crick complement: A-U, U-A, C-G, or G-C.

[No sharp turns.] The ends of each pair are separated by at least 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 $i < k < j < l$.

Free energy. Usual hypothesis is that an RNA molecule will form the secondary structure with the optimum total free energy.

approximate by number of base pairs

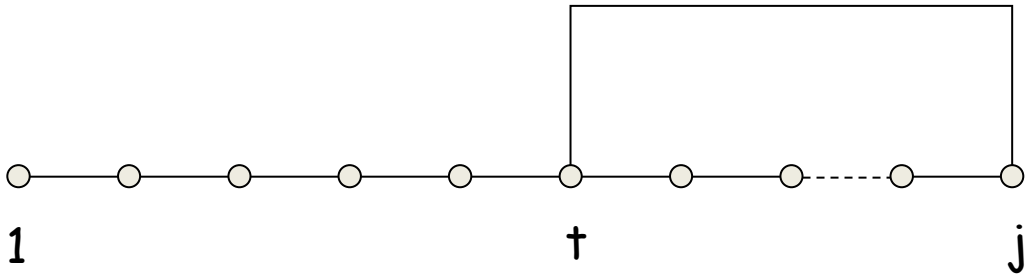
Goal. Given an RNA molecule $B = b_1 b_2 \ldots b_n$, find a secondary structure S that maximizes the number of base pairs.

RNA Secondary Structure: Subproblems

First attempt. OPT(j) = maximum number of base pairs in a secondary structure of the substring $b_1 b_2 \ldots b_j$.

Case 1: b_j is not involved in a pair : OPT(j)=OPT(j-1)

Case 2: b_j matches b_t for some $1 \le t < j-4$



Difficulty. Results in two sub-problems:

Finding secondary structure in: $b_1 b_2 \ldots b_{t-1}$ ← OPT(t-1)

Finding secondary structure in: $b_{t+1} b_{t+2} \ldots b_{j-1}$ ← need different sub-problems

Dynamic Programming Over Intervals

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

Case 1. If $i \geq j - 4$.

OPT(i, j) = 0 by no-sharp turns condition.

Case 2. Base b_j is not involved in a pair.

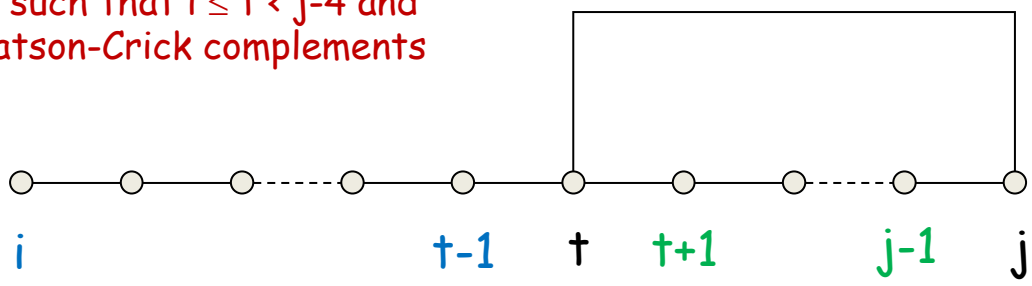
OPT(i, j) = OPT(i, j-1)

Case 3. Base b_j pairs with b_t for some $i \leq t < j - 4$.

non-crossing constraint (no match over t) decouples resulting sub-problems

$OPT(i, j) = \max_t \{1 + OPT(i, t-1) + OPT(t+1, j-1)\}$

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



Relazione di ricorrenza

OPT(i, j) = maximum number of base pairs in a secondary structure of the substring $b_i b_{i+1} \ldots b_j$ with i, j = 1, 2, … , n

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & i \geq j-4 \\ \max \begin{cases} \mathrm{OPT}(i, j-1) \\ \max\limits_{\substack{i \leq t < j-4 \\ b_j b_t \text{ complementi}}} \{ 1 + \mathrm{OPT}(i, t-1) + \mathrm{OPT}(t+1, j-1) \} \end{cases} & \text{altrimenti} \end{cases}$$

Dynamic Programming Over Intervals

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

Case 1. If $i \geq j - 4$.

OPT(i, j) = 0 by no-sharp turns condition.

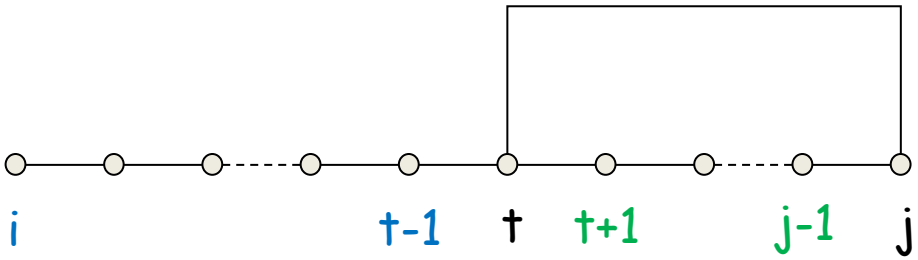
Case 2. Base b_j is not involved in a pair.

OPT(i, j) = OPT(i, j-1)

Case 3. Base b_j pairs with b_t for some $i \leq t < j - 4$.

non-crossing constraint (no match over t) decouples resulting sub-problems

$OPT(i, j) = 1 + \max_t \{ OPT(i, t-1) + OPT(t+1, j-1) \}$



Un esempio

Supponiamo che la stringa in ingresso sia

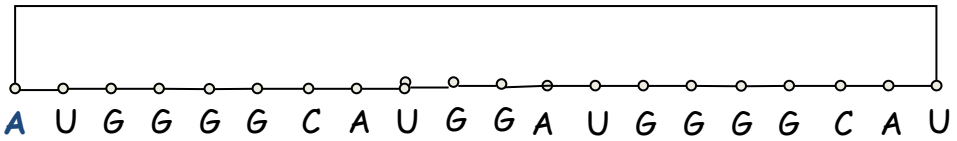
$$b_1 b_2 \ldots b_{20} = AUGGGGCAUGGAUGGGGCAU.$$

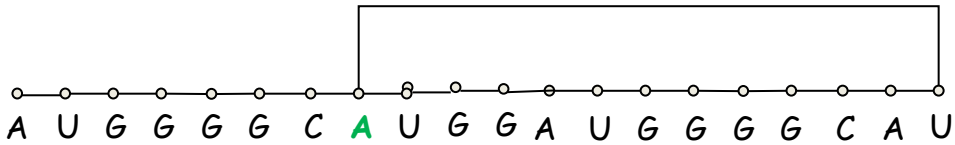
Allora

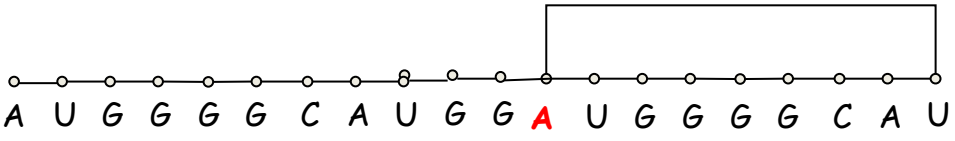
$$OPT(1,20) = \max \begin{cases} OPT[1,19] \\ \max \begin{cases} 1 + OPT(1,0) + OPT(2,19) \\ 1 + OPT(1,7) + OPT(9,19) \\ 1 + OPT(1,11) + OPT\ (13,19) \end{cases} \end{cases}$$

in quanto vi sono tre valori di t per cui $1 \le t < 16$ e b_t =A è complementare a b_{20} = U.

Sono i valori t = 1, 8, 12.







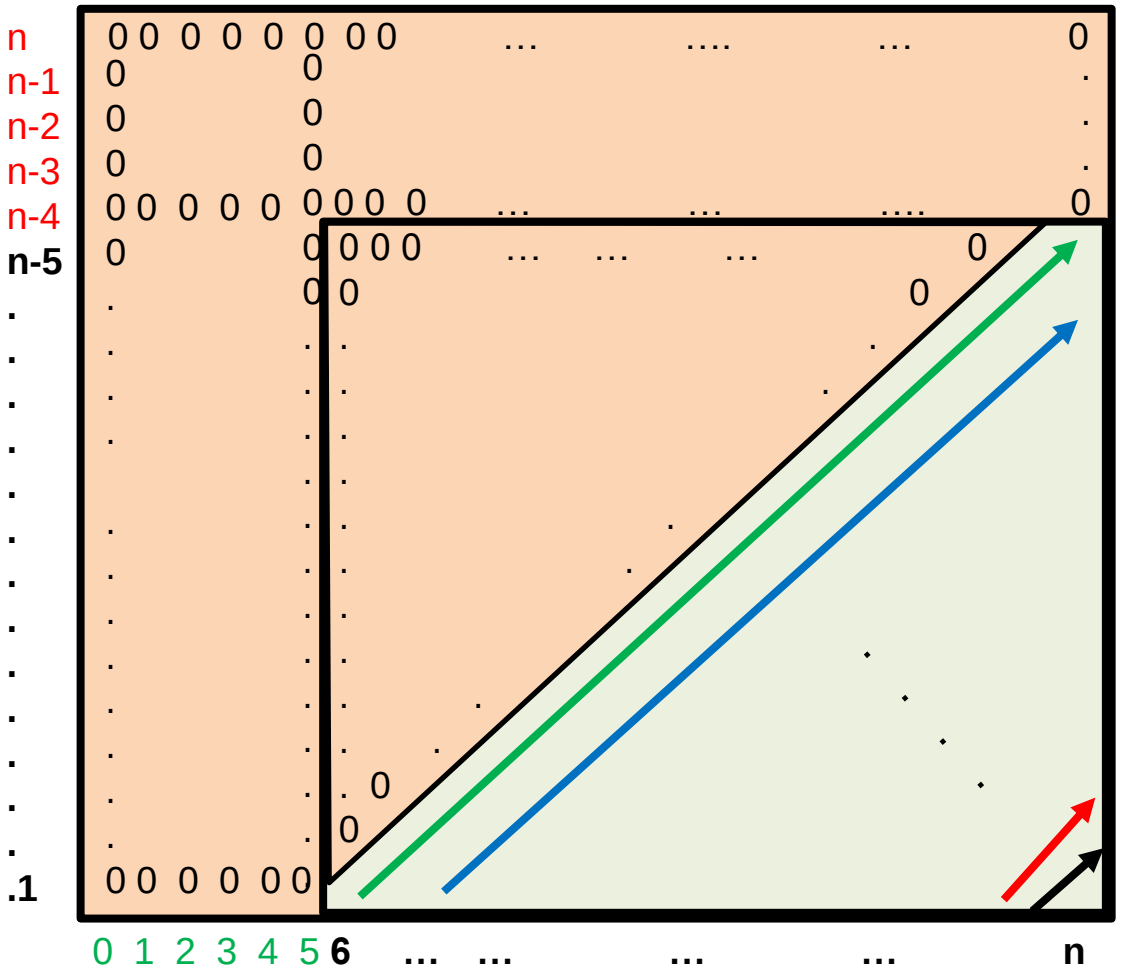
La tabella: casi base

OPT(i, j) = maximum number of base pairs in a secondary structure of the substring $b_i b_{i+1} \ldots b_j$ with i = 1, 2, … , n, j = 0,1, 2, … , n

OPT(i, j) = 0 se i ≥ j-4

Le righe i = n – 4, … , n
e le colonne j = 0, … , 5
contengono 0.

Spazio S(n) = Θ (n^2)



La tabella: caso generale

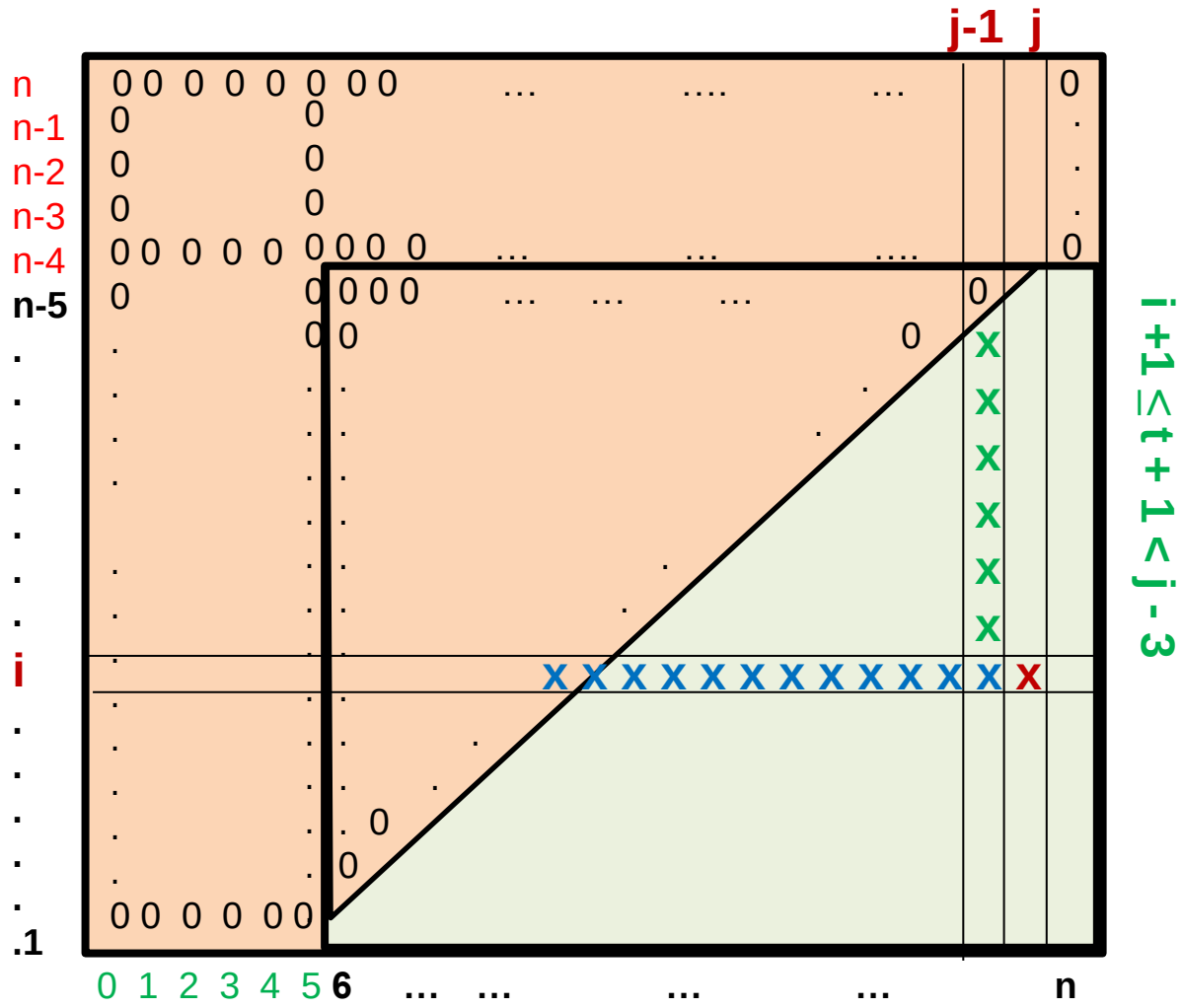
$$OPT(i, j) = \max \begin{cases} OPT(i, j-1) \\ \max \{ 1 + OPT(i, t-1) + OPT(t+1, j-1) \} \end{cases}$$

(*)

(*)
$i \leq t < j-4$
b_t e b_j complementari

$i - 1 \leq t - 1 < j - 5$

La parte verde della tabella sarà riempita per diagonali, partendo da quella sotto la diagonale di 0



Relazione di ricorrenza

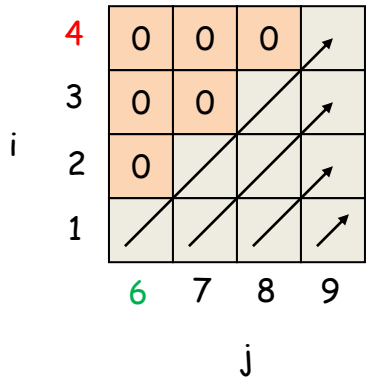
OPT(i, j) = maximum number of base pairs in a secondary structure of the substring $b_i b_{i+1} \ldots b_j$ with i = 1, 2, … , n, j = 0,1, 2, … , n

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & i \geq j-4 \\ \max \begin{cases} \mathrm{OPT}(i,j-1) \\ \max\limits_{\substack{i \leq t < j-4 \\ b_j b_t \text{ complementi}}} \{\, 1 + \mathrm{OPT}(i,t-1) + \mathrm{OPT}(t+1,j-1) \,\} \end{cases} & \text{altrimenti} \end{cases}$$

Caso generale: i< j – 4 ovvero i ≤ j-5, ovvero j ≥ i+5

i = 1 allora j ≥ 6
i = 2 allora j ≥ 7
…
i = n-5 allora j ≥ n
i = n-4 allora j ≥ n+1
…
i = n allora j ≥ n+5

Esempio: n = 9 allora
i ≤ 9-5 = 4, j ≥ 6



Bottom Up Dynamic Programming Over Intervals

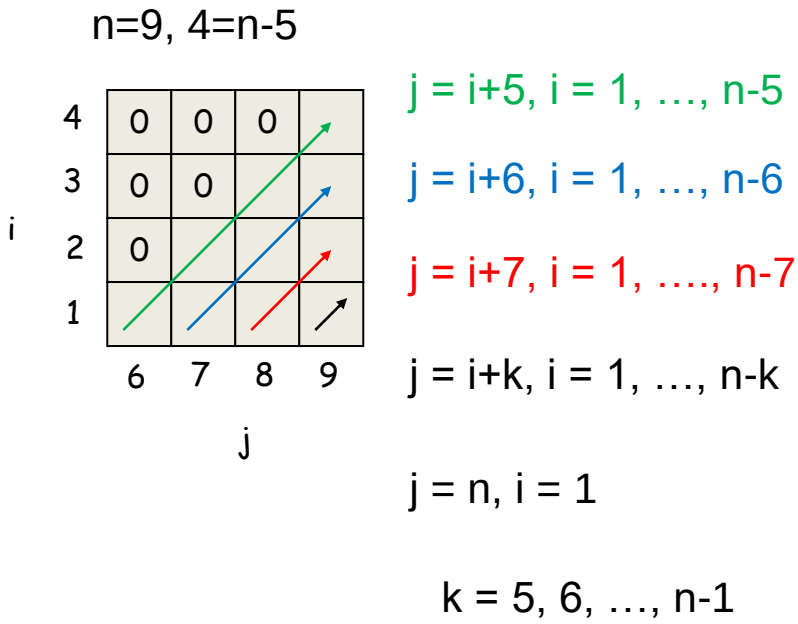
Q. What order to solve the sub-problems?

A. Do shortest intervals first (from length 5 up to n-1).

```
RNA(b_1,…,b_n) {
    for k = 5, 6, …, n-1
        for i = 1, 2, …, n-k
            j = i + k
            Compute M[i, j]

    return M[1, n]
}
```

using recurrence



Running time. $O(n^3)$.

Esempio

RNA sequence *ACCGGUAGU*

4	0	0	0	
3	0	0		
2	0			
$i = 1$				
	$j = 6$	7	8	9

Initial values

4	0	0	0	0
3	0	0	1	
2	0	0		
$i = 1$	1			
	$j = 6$	7	8	9

Filling in the values for $k = 5$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	
$i = 1$	1	1		
	$j = 6$	7	8	9

Filling in the values for $k = 6$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
$i = 1$	1	1	1	
	$j = 6$	7	8	9

Filling in the values for $k = 7$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
$i = 1$	1	1	1	2
	$j = 6$	7	8	9

Filling in the values for $k = 8$

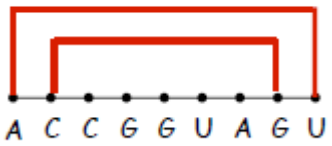
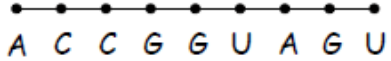


Figure 6.16 The iterations of the algorithm on a sample instance of the RNA Secondary Structure Prediction Problem.

RNA sequence *ACCGGUAGU*



4	0	0	0	
3	0	0		
2	0			
$i = 1$				
	$j = 6$	7	8	9

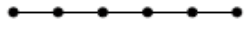
Initial values

4	0	0	0	0
3	0	0	1	
2	0	0		
$i = 1$	1			
	$j = 6$	7	8	9

Filling in the values for $k = 5$

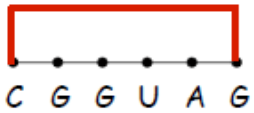
$$OPT(i, j) = \max \begin{cases} OPT(i, j-1) \\ \max_{\substack{i \le t < j-4 \\ b_j b_t \text{ complementi}}} \{ 1 + OPT(i, t-1) + OPT(t+1, j-1) \} \end{cases}$$

$$OPT(4, 9) = \max \begin{cases} OPT(4, 8) = 0 \\ \max_{\substack{4 \le t < 5 \\ b_9 b_t \text{ complementi}}} \{ 1 + OPT(4, t-1) + OPT(t+1, 8) \} = 0 \end{cases}$$

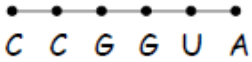


b_4=*G* does not match b_9=U

$$OPT(3, 8) = \max \begin{cases} OPT(3, 7) = 0 \\ \max_{\substack{3 \le t < 4 \\ b_8 b_t \text{ complementi}}} \{1 + OPT(3, t-1) + OPT(t+1, 7) \} = 1 \end{cases}$$

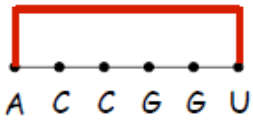


$$OPT(2, 7) = \max \begin{cases} OPT(2, 6) = 0 \\ \max_{\substack{2 \le t < 3 \\ b_7 b_t \text{ complementi}}} \{ 1 + OPT(2, t-1) + OPT(t+1, 6) \} = 0 \end{cases}$$



b_2=*C* does not match b_7=*A*

$$OPT(1, 6) = \max \begin{cases} OPT(1, 5) = 0 \\ \max_{\substack{1 \le t < 2 \\ b_6 b_t \text{ complementi}}} \{ 1 + OPT(1, t-1) + OPT(t+1, 5) \} = 1 \end{cases}$$



RNA sequence *ACCGGUAGU*

4	0	0	0	
3	0	0		
2	0			
= 1				
	j = 6	7	8	9

Initial values

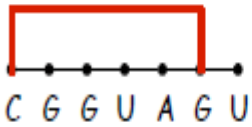
4	0	0	0	0
3	0	0	1	
2	0	0		
i = 1	1			
	j = 6	7	8	9

Filling in the values for $k = 5$

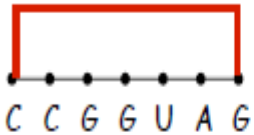
4	0	0	0	0
3	0	0	1	1
2	0	0	1	
i = 1	1	1		
	j = 6	7	8	9

Filling in the values for $k = 6$

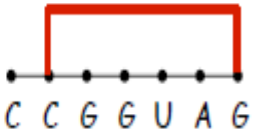
$$OPT(3,9) = \max \begin{cases} OPT(3,8) = 1 \\ \max\limits_{\substack{3 \le t < 5 \\ b_9 = U, b_t \ compl.}} \{1 + OPT(3, t-1) + OPT(t+1, 8)\} = 0 \end{cases}$$



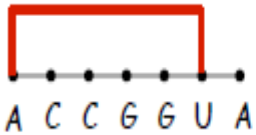
$$OPT(2,8) = \max \begin{cases} OPT(2,7) = 0 \\ \max\limits_{\substack{2 \le t < 4 \\ b_8 = G, b_t \ compl.}} \{1 + OPT(2, t-1) + OPT(t+1, 7)\} = 1 \end{cases}$$



oppure



$$OPT(1,7) = \max \begin{cases} OPT(1,6) = 1 \\ \max\limits_{\substack{1 \le t < 3 \\ b_7 = A, b_t \ compl.}} \{1 + OPT(1, t-1) + OPT(t+1, 6)\} = 0 \end{cases}$$



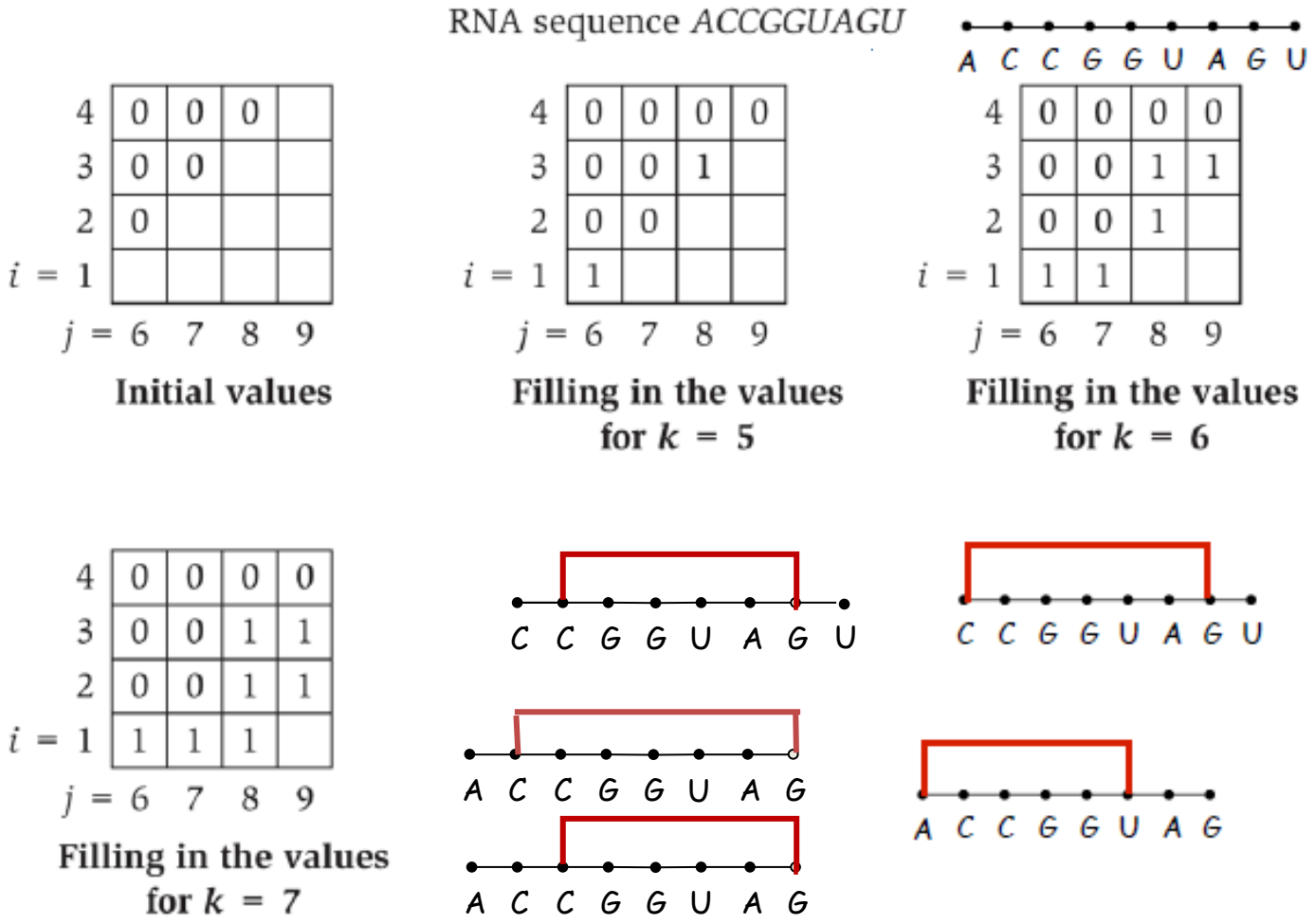


Figure 6.16 The iterations of the algorithm on a sample instance of the RNA Secondary Structure Prediction Problem.

RNA sequence *ACCGGUAGU*

i	$j=6$	7	8	9
4	0	0	0	
3	0	0		
2	0			
1				

Initial values

i	$j=6$	7	8	9
4	0	0	0	0
3	0	0	1	
2	0	0		
1	1			

Filling in the values for $k = 5$

i	$j=6$	7	8	9
4	0	0	0	0
3	0	0	1	1
2	0	0	1	
1	1	1		

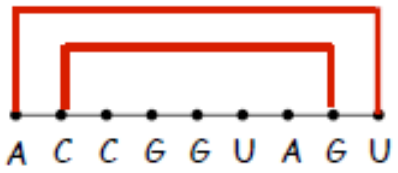
Filling in the values for $k = 6$

i	$j=6$	7	8	9
4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
1	1	1	1	

Filling in the values for $k = 7$

i	$j=6$	7	8	9
4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
1	1	1	1	2

Filling in the values for $k = 8$



oppure

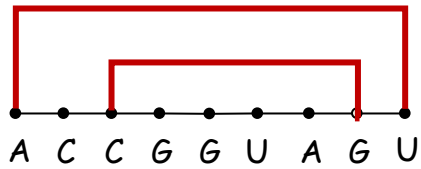


Figure 6.16 The iterations of the algorithm on a sample instance of the RNA Secondary Structure Prediction Problem.

Trovare una soluzione

RNA sequence *ACCGGUAGU*

4	0	0	0	
3	0	0		
2	0			
$i = 1$				
	$j = 6$	7	8	9

Initial values

4	0	0	0	0
3	0	0	1	
2	0	0		
$i = 1$	1			
	$j = 6$	7	8	9

Filling in the values for $k = 5$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	
$i = 1$	1	1		
	$j = 6$	7	8	9

Filling in the values for $k = 6$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
$i = 1$	1	1	1	
	$j = 6$	7	8	9

Filling in the values for $k = 7$

4	0	0	0	0
3	0	0	1	1
2	0	0	1	1
$i = 1$	1	1	1	2
	$j = 6$	7	8	9

Filling in the values for $k = 8$

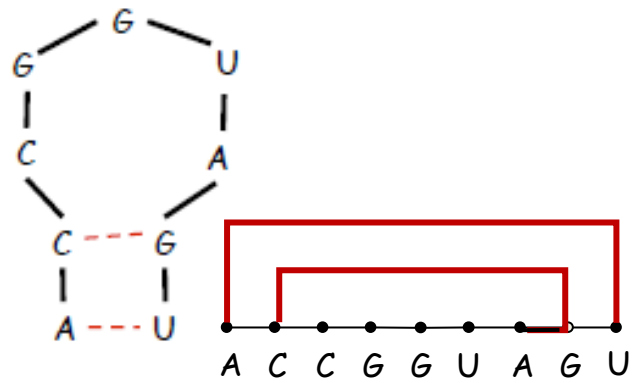


Figure 6.16 The iterations of the algorithm on a sample instance of the RNA Secondary Structure Prediction Problem.

Dynamic Programming Summary

Recipe

- Characterize structure of problem.
- Recursively define value of optimal solution.
- Compute value of optimal solution.
- Construct optimal solution from computed information.

Dynamic programming techniques

- Binary choice: weighted interval scheduling, sequence alignment
- Adding a new variable: knapsack.
- Dynamic programming over intervals: RNA secondary structure.
- Multi-way choice: Bellman-Ford's algorithm for shortest paths in a weighted directed graph

(vedremo dapprima una versione semplificata: *esercizio della canoa*)

Top-down vs. bottom-up: different people have different intuitions.