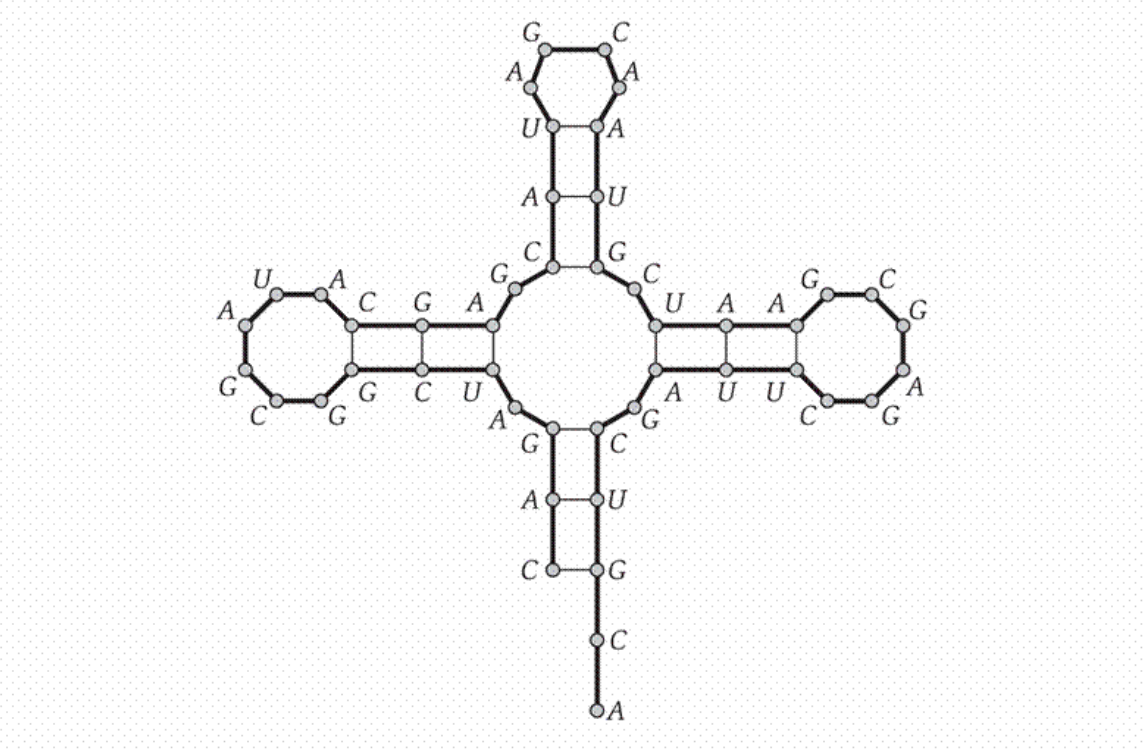
Finding RNA Secondary Structures



An RNA molecule with n bases:

- $b_1\ b_2\ \ldots\ b_n$
- each b_i is one of {A, C, G, U}

→

Strings formed by characters from the alphabet {A, C, G, U}

Examples of RNA strings

ACCGGUAGU

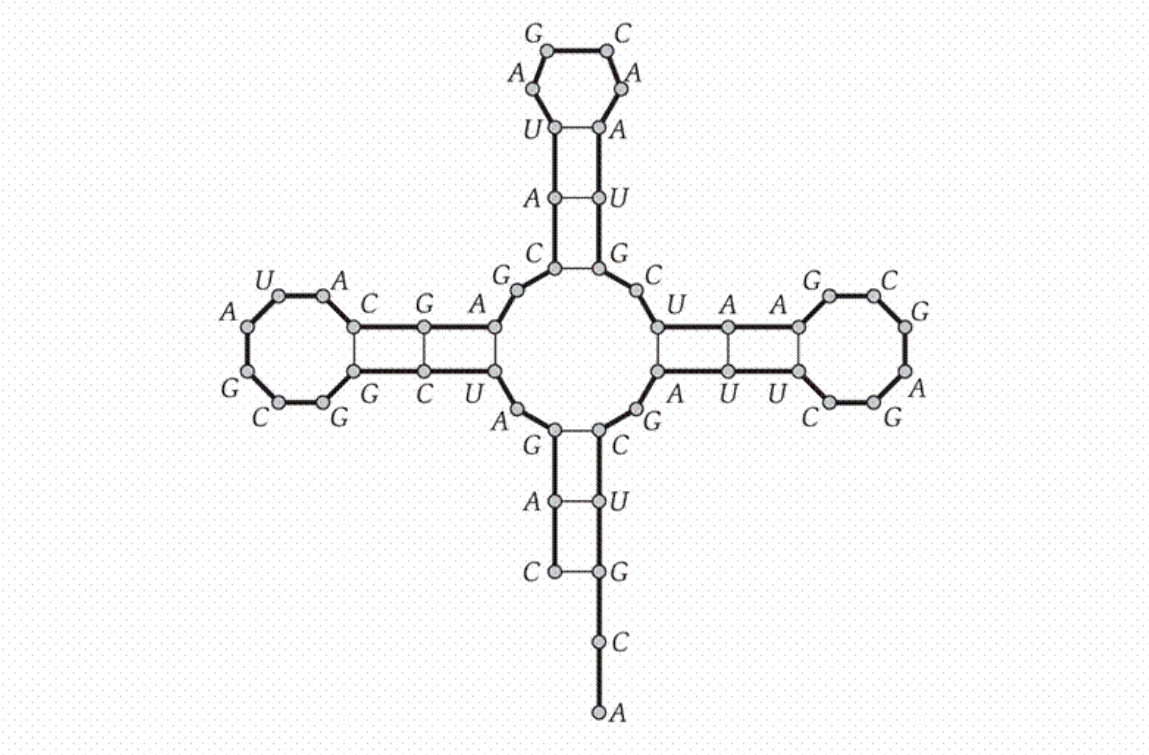
ACAUGAUGGCCAUGU

ACGUGCGAUUCGAGCGAAUCGUAACGAUACGAGCAUAGCGGCUAGAC

RNA Secondary structures

- A RNA molecule loops back to form base pairs with itself

ACGUCGAUUCGAGCGAAUCGUAACGAUACGAGCAUAGCGGCUAGAC



Constraints of RNA Secondary Structures

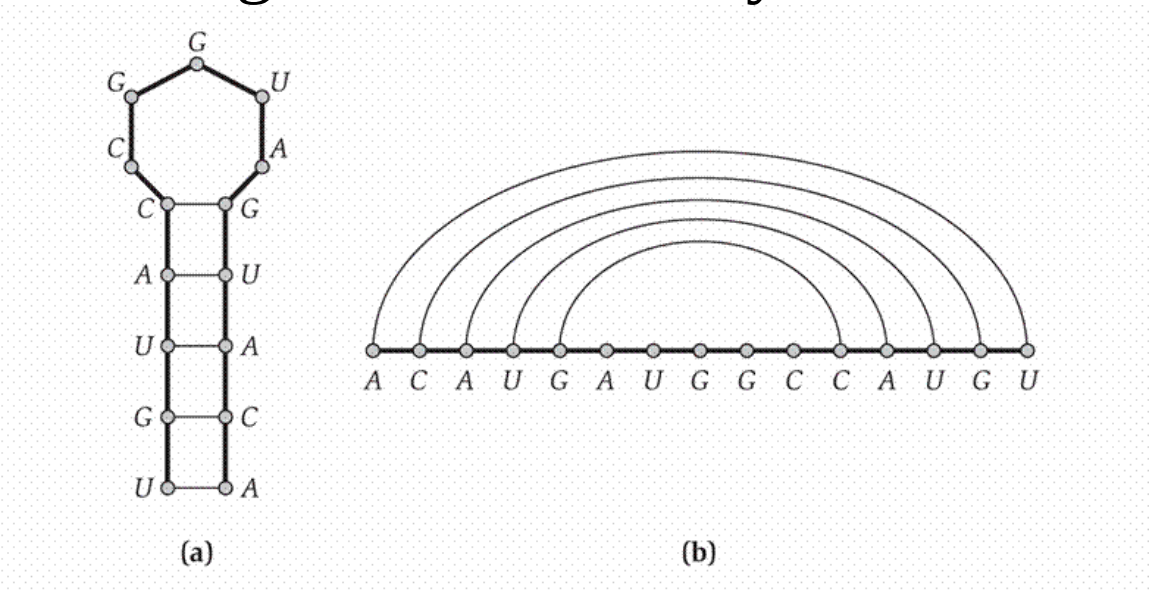
- Matching A with U, C with G
- Each base is matched at most once
- No near-neighbor matching:
  No sharp turn
- No crossing between 2 matched pairs

Constraints of RNA Secondary Structures

Let $i < j$, a matched pair (b_i, b_j) →

- (b_i, b_j) is one of
  { (A,U) , (U,A) , (C,G), (G,C) }
- None of b_i , b_j can be in other pairs
- $i < j - 4$
- Let $x < y$, no (b_x, b_y) pairs allowed
  where $x < i < y < j$
  or $i < x < j < y$

Finding RNA Secondary Structures



More matched pairs → More likely a RNA secondary

Finding most likely secondary structures
→
Finding maximum matching
with respect to the constraints

Finding Maximum Matching in RNA

- For all valid indices i and j where $1 \leq i < j \leq n$,

  let OPT(i, j) be :

  The maximum number of pairs we can form in the segment of RNA molecule from the i-th base to the j-th base

Algorithm for maximum matching

- Boundary conditions:

  when $j - i \leq 4$

  $OPT(i, j) = 0$

- Recurrence equation:

  when $j - i > 4$

  $OPT(i, j) =$

  $Max\ (\ OPT(i, j-1),$

  $Max_t\ (1 + OPT(i, t-1) +$

  $OPT(t+1, j-1)$

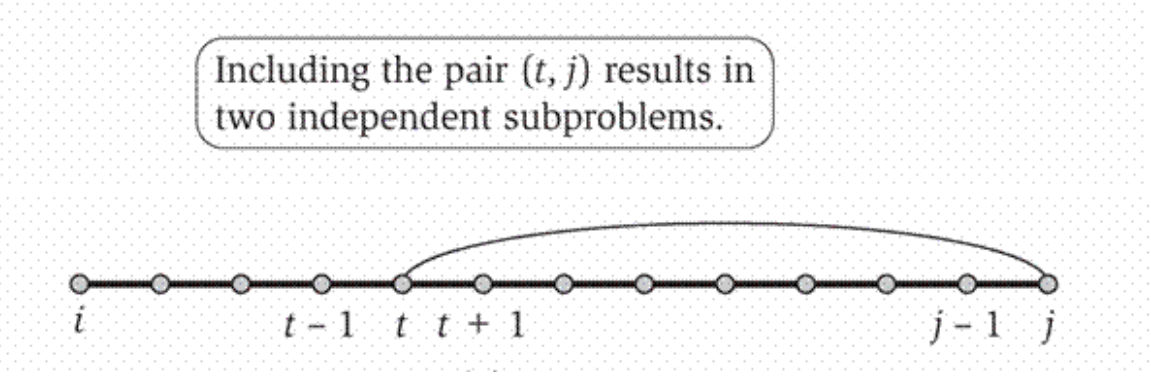
  $)$

  $)$

  where $i \leq t < j$ and

  (b_t, b_j) can form a pair

Algorithm for maximum matching



Recurrence equation:

OPT(i, j) =

Max (OPT(i, j-1),

Max $_t$ (1 + OPT(i, t-1)+

OPT(t+1,j-1)

)

)

Algorithm for maximum matching

- Apply dynamic programming to systematically calculate OPT(i, j) for $1 \leq i < j \leq n$,

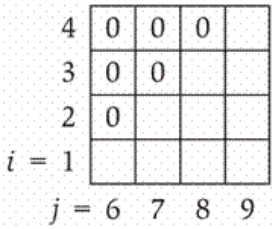
- Trace the results of OPT(i, j) values to find the most likely RNA secondary structures

Calculate All OPT(i, j) Values: Dynamic programming

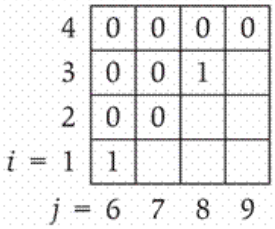
```
Initialize OPT(i, j) = 0 whenever i ≥ j − 4
For k = 5, 6, . . . , n − 1
  For i = 1, 2, . . . n − k
    Set j = i + k
    Compute OPT(i, j) using the recurrence
  Endfor
Endfor
Return OPT(1, n)
```

Calculate All OPT(i, j) Values: Dynamic programming

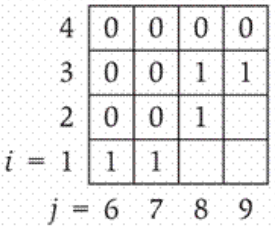
RNA sequence *ACCGGUAGU*



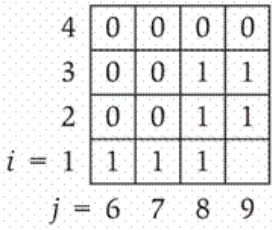
Initial values



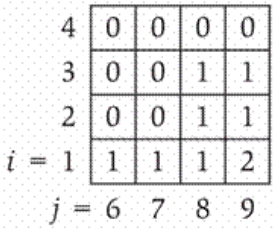
Filling in the values for $k = 5$



Filling in the values for $k = 6$



Filling in the values for $k = 7$



Filling in the values for $k = 8$

Complexity of Finding RNA Secondary Structures

- Given a RNA sequence of n bases, $O(n^2)$ subproblems to solve to determine OPT(i, j) $1 \leq i < j \leq n$
- Using the recurrence & dynamic programming, each subproblem solved in O(n) time
- O(n) time to trace back to find each most likely secondary structure
- Let k be the number of most likely secondary structures
- It takes $O(n^3+kn)$ time to solve it