

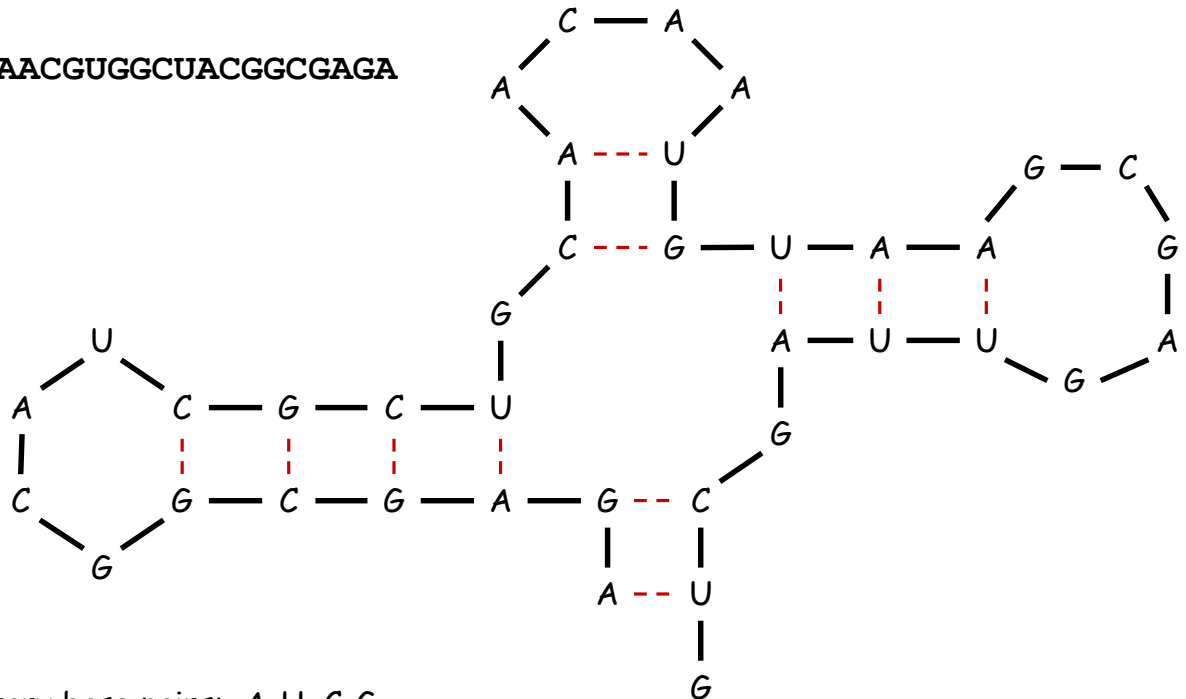
RNA Secondary Structure

RNA Secondary Structure

RNA. String $B = b_1b_2\dots b_n$ over alphabet $\{A, C, G, U\}$.

Secondary structure. RNA is single-stranded so it tends to loop back and form base pairs with itself. This structure is essential for understanding behavior of molecule.

Ex: GUCGAUUGAGCGAUGUAACAACGUGGCUACGGCGAGA



complementary base pairs: A-U, C-G

RNA Secondary Structure

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

- [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 $i < k < j < l$.

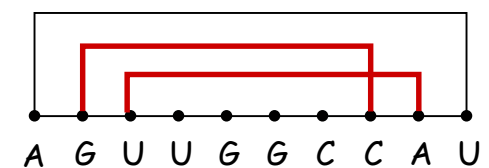
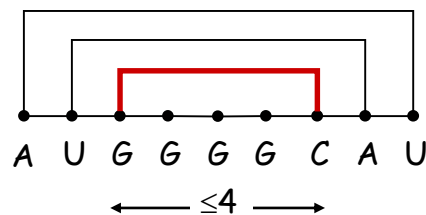
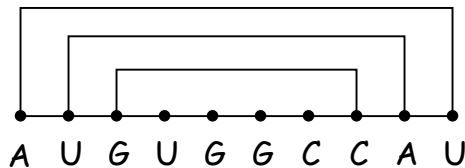
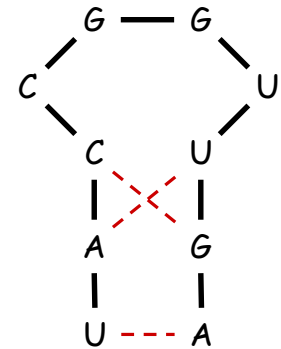
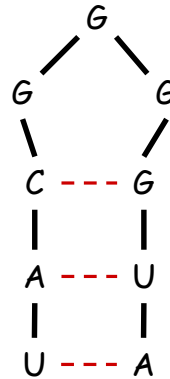
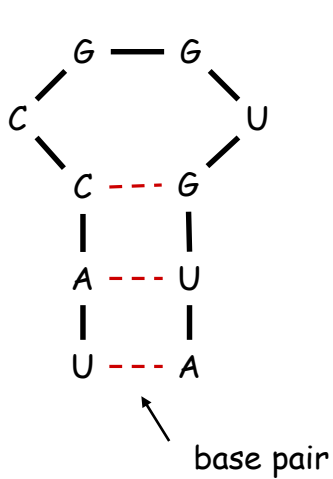
Free energy. Usual hypothesis is that an RNA molecule will form the secondary structure with the optimum total free energy.

↑
approximate by number of base pairs

Goal. Given an RNA molecule $B = b_1b_2\dots b_n$, find a secondary structure S that maximizes the number of base pairs.

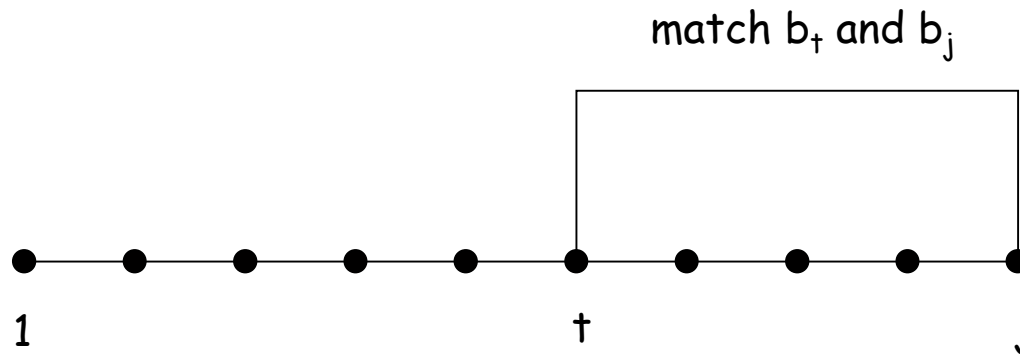
RNA Secondary Structure: Examples

Examples.



RNA Secondary Structure: Subproblems

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



Difficulty. Results in two sub-problems.

- Finding secondary structure in: $b_1b_2\dots b_{t-1}$. $\leftarrow \text{OPT}(t-1)$
- Finding secondary structure in: $b_{t+1}b_{t+2}\dots b_{j-1}$. \leftarrow need more sub-problems

Dynamic Programming Over Intervals

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

- Case 1. If $i \geq j - 4$.
 - $\text{OPT}(i, j) = 0$ by no-sharp turns condition.
- Case 2. Base b_j is not involved in a pair.
 - $\text{OPT}(i, j) = \text{OPT}(i, j-1)$
- Case 3. Base b_j pairs with b_t for some $i \leq t < j - 4$.
 - non-crossing constraint decouples resulting sub-problems
 - $\text{OPT}(i, j) = 1 + \max_t \{ \text{OPT}(i, t-1) + \text{OPT}(t+1, j-1) \}$

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

Remark. Same core idea in CKY (Cocke-Younger-Kasami) algorithm to parse context-free grammars.

Bottom Up Dynamic Programming Over Intervals

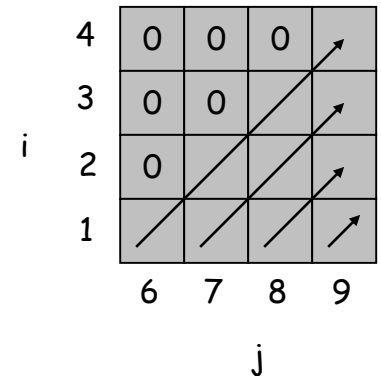
Q. What order to solve the sub-problems?

A. Do shortest intervals first.

```
RNA( $b_1, \dots, b_n$ ) {  
  for  $k = 5, 6, \dots, n-1$   
    for  $i = 1, 2, \dots, n-k$   
       $j = i + k$   
       $M[i, j]$   
        =  $\max\{M[i, j-1], 1 + \max_{i \leq t < j-4, b_t \text{ compatible with } b_j} \{M[i, t-1] + M[t+1, j-1]\}\}$   
  return  $M[1, n]$   
}
```

↖
using recurrence

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



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.

- Adding a new variable: knapsack.
- Binary/multi-way choice: weighted interval scheduling.
- Dynamic programming over intervals: RNA secondary structure.

↖ CKY parsing algorithm for context-free grammar has similar structure