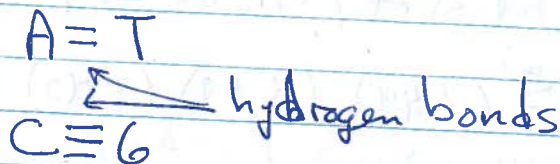
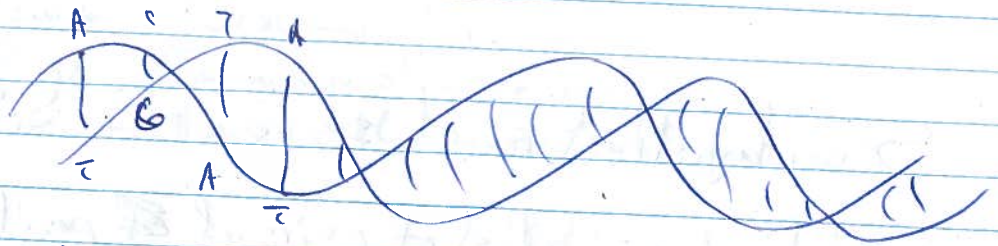
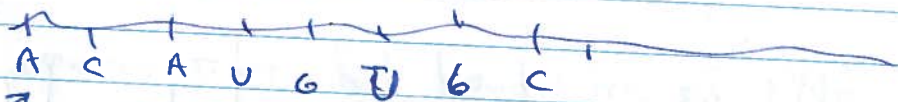


RNA secondary structure prediction

DNA
Stable:



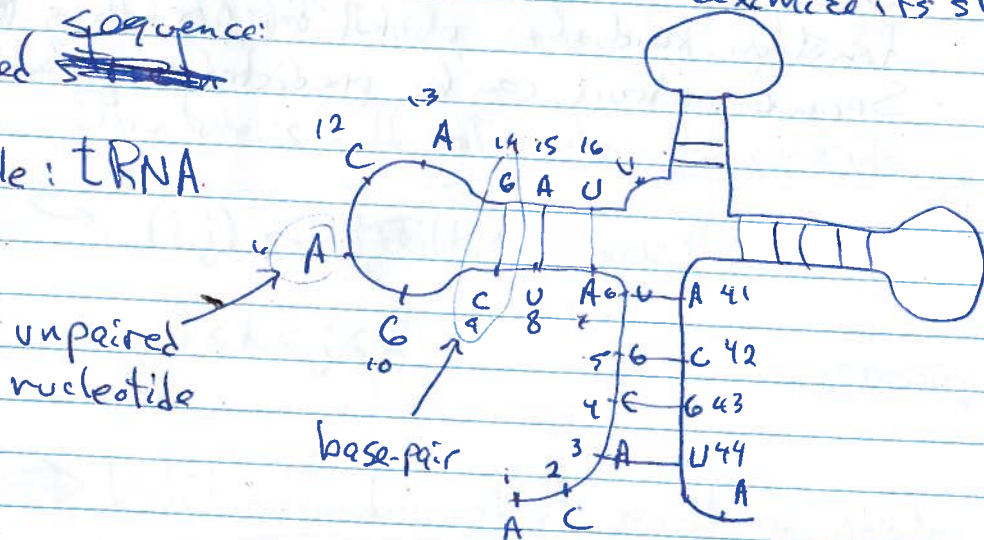
RNA: single-stranded chain of nucleotide



Unfolded structure is unstable \Rightarrow Sequence folds to maximize its stability

Folded ~~sequence~~

Example: tRNA



Function of RNA molecules depends on their structure
 \swarrow
depends on sequence

Key ideas: A sequence typically folds in its most stable structure

Secondary structure of sequence $S = S_1 S_2 \dots S_L$

is defined as list of pairs of positions that form base pairs

Sec. struct: $\{(3, 44), (4, 43), (5, 42), (6, 41), (7, 40), (8, 39), \dots\}$

Tertiary structure \equiv 3D structure: (x, y, z) coordinates of each atom in the sequence

Idea: We want tertiary structure, but it is hard to predict

- Secondary struct. is sufficient to let us predict function, and also useful for predicting tertiary struct
- Secondary struct. can be predicted computationally

Secondary Structure prediction problem


Version 1: Given: Sequence $S_1 \dots S_n$

Find: Secondary structure for $S_1 \dots S_n$
that is the most stable

↓
of base pairings present
in the sec. struct.

Example:

1 2 3 4 5 6 7 8 9 10 11 12 13 14
A C A C G U A U C G U U A C



$\{(1,6), (2,5), (3,8), \dots\}$

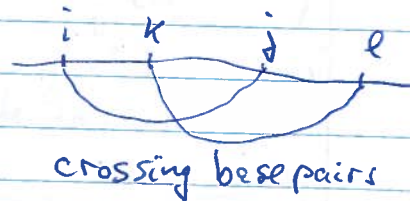
Problem: Ignores rules about bendability of RNA

Rules

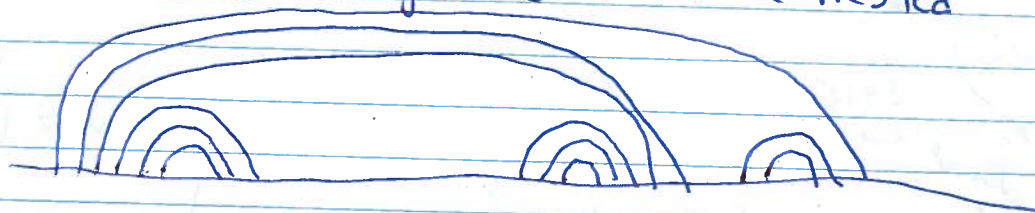
① if $(i,j) \in \text{Struct}$, then $|i-j| \geq 3$

② Structure should not contain pseudoknots

→ (i,j) and (k,l) such that
 $i < k < j < l$



⇒ Valid secondary struct must be nested



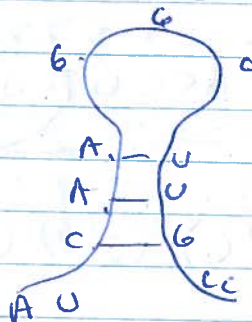
Version 2: Given = RNA seq S_1 and S_2

Find = Sec. struct. that maximizes total # of base pairs, subject to ~~the~~ rules ① and ②

Example:

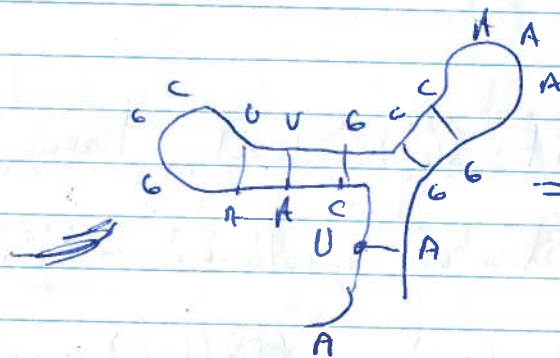
A U C A A G G C U U G C C | A A A G G A

Solution:
(short)



\Rightarrow 3 base pairs

longer:



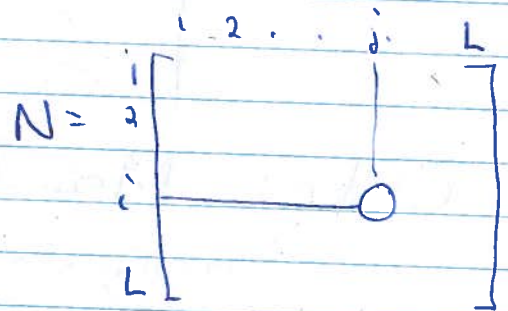
\Rightarrow 6 base pairs

Nussinov Algorithm: Dynamic prog. algo.

Define $N(i, j)$ = Maximum # of base pairs that can be formed for $S_i \dots S_j$

We want $N(1, L)$

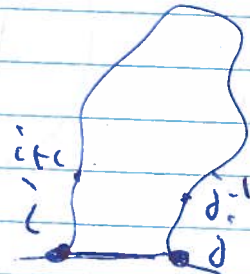
$S = 1 \ 2 \ 3 \ i \ \dots \ j \ L$



How to calculate $N(i, j)$?

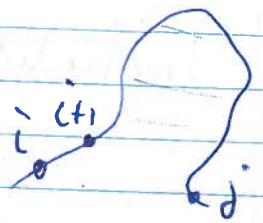
Case 1: S_i is paired with S_j

$$1 + N(i+1, j-1)$$



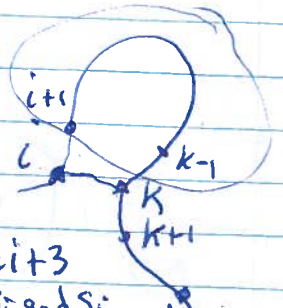
Case 2: S_i is not paired with anything

$$0 + N(i+1, j)$$



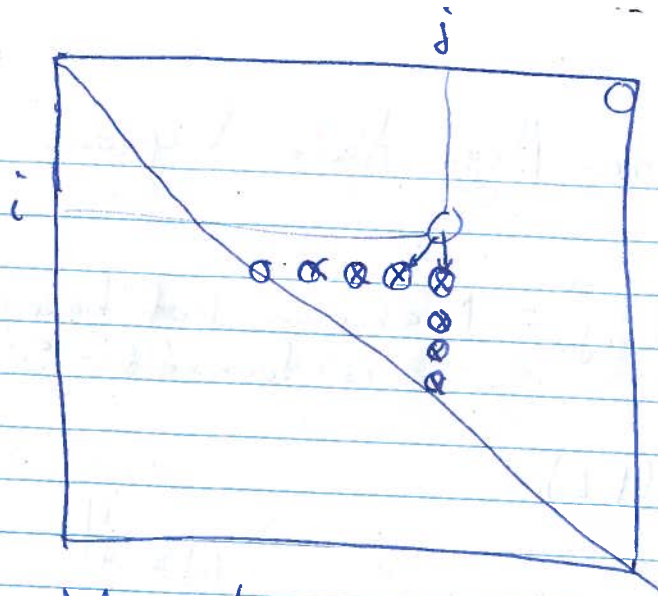
Case 3: S_i is paired with some nucleotide S_k , where $k < j$

$$1 + N(i+1, k-1) + N(k+1, j)$$

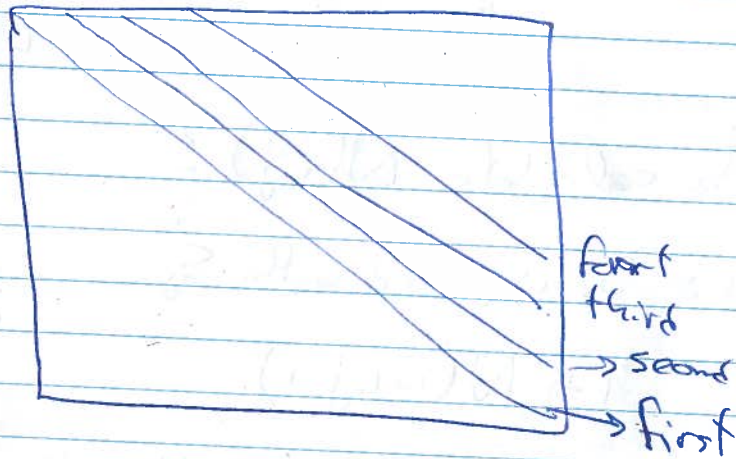


$$N(i, j) = \max \begin{cases} 1 + N(i+1, j-1) \leftarrow \text{if } j \geq i+3 \text{ and } S_i \text{ and } S_j \text{ are complementary} \\ 0 + N(i+1, j) \\ 1 + \max \{ N(i+1, k-1) + N(k+1, j) \} \\ \quad i+3 \leq k < j, \text{ and } S_i \text{ is complementary to } S_k \end{cases}$$

N:



Order: Main diagonal \rightarrow Corner



Initialization: $N(i, i) = 0 \quad \forall i \leq L$
 $N(i, i+1) = 0$
 $N(i, i+2) = 0$

From there, use recurrence