RNA Folding

RNA: ACGGGGGUUUUAAAUUUCCCCUAUAT

In the cell, RNA folds up:

- G and C stick together
- A and U stick together
RNA Fold Example
RNA Folding Rules:

1. If two bases are closer than 4 bases apart, they cannot pair.
2. Each base is matched to at most one other base.
3. The allowable pairs are \{U, A\} and \{G, C\}.
4. Pairs cannot “cross.”
No Crossings

This is not allowed:

In other words, if \((i, j)\) and \((k, \ell)\) are paired, then we must have \(i < k < \ell < j\).

Paired bases have to be nested.
RNA Folding Problem

RNA Folding

Given: a string $r = b_1, b_2, b_3, \ldots, b_n$, with $b_i \in \{A, C, U, G\}$

Find: the largest set of pairs $S = \{(i, j)\}$, where $i, j \in \{1, 2, \ldots, n\}$, that satisfies the RNA Folding Rules.

Goal: match as many pairs of bases as possible.
Subproblems

\[ j \text{ is not paired with anything} \]

\[ j \text{ is paired with some } t \leq j - 4 \]

\[ \text{OPT}(1, j-1) \]

\[ \text{OPT}(1, t-1) \]

\[ \text{OPT}(t+1, j-1) \]
We have a subproblem for every interval \((i, j)\).

How many subproblems are there?
We have a subproblem for every interval \((i, j)\).

How many subproblems are there?

\[
\binom{n}{2} = O(n^2)
\]
Recurrence

If \( j - i \leq 4 \): \( OPT(i, j) = 0 \)

If \( j - i > 4 \):

\[
OPT(i, j) = \max \left\{ OPT(i, j - 1), \max_t \{ 1 + OPT(i, t - 1) + OPT(t + 1, j - 1) \} \right\}
\]

In the 2nd case above, we try all possible \( t \) to pair with \( j \). That is \( t \) runs from \( i \) to \( j - 4 \).
• In what order should we solve the subproblems?

• What problems do we need to solve $OPT(i, j)$?
Order to Solve the Problems

- In what order should we solve the subproblems?

- What problems do we need to solve $OPT(i,j)$?

- $OPT(i, t - 1)$ and $OPT(t + 1, j - 1)$ for every $t$ between $i$ and $j$.

- In what sense are these problems “smaller?”
Order to Solve the Problems

- In what order should we solve the subproblems?
- What problems do we need to solve $OPT(i, j)$?
  - $OPT(i, t - 1)$ and $OPT(t + 1, j - 1)$ for every $t$ between $i$ and $j$.
- In what sense are these problems “smaller?”
- They involve smaller intervals of the string.

**Subproblem Ordering**

We solve $OPT(i, j)$ in order of increasing value of $j - i$. 
Initialize $\text{OPT}[i,j]$ to 0 for $1 \leq i,j \leq n$

For $k = 5, 6, \ldots, n-1$ // interval length
    For $i = 1, 2, \ldots, n-k$ // interval start
        Set $j = i + k$ // interval end

        // find the best $t$
        best_t = 0
        For $t = i, \ldots, j-1$:
            best_t = max(best_t, 1 + $\text{OPT}[i,t-1]$+$\text{OPT}[t+1,j-1]$)

        // Either pair $j$ with $t$ or nothing
        $\text{OPT}[i,j] = \max(\text{best}_t, \text{OPT}[i,j-1])$
    EndFor
EndFor
Return $\text{OPT}[1,n]$
1. $O(n^2)$ subproblems.

2. Now, it takes $O(n)$ time to solve each subproblem. (have to search over all possible choices for $t$.)

3. Total running time is $O(n^3)$. 