2/18

Announcement : Prof. Kleinberg is holding extra office hours today from 1:30 pm to 2:30pm . [Gates 317]

RNA secondary structure.

RNA molecule: sequence of symbols/bases.

Symbols/bases: $\{A, G, C, U\}$.

A strand of length n :

$x_1\, x_2 \; \cdots \; x_n$, each x_i is a base.

Secondary structure with a strand.

$S \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$

if $(i,j) \in S$ then $i < j$

(i) No sharp turns: if $(i,j) \in S$ then $i < j-4$.

(ii) if $(i,j) \in S$ then $\{x_i, x_j\}$ is either $\{A, U\}$ or $\{G, C\}$. [valid pairs]

(iii) S to be a matching.

(iv) (No crossing) (i_1, j_1) and (i_2, j_2) are in S,

suppose $i_1 < i_2$, then

either (a) $j_1 < i_2$ or

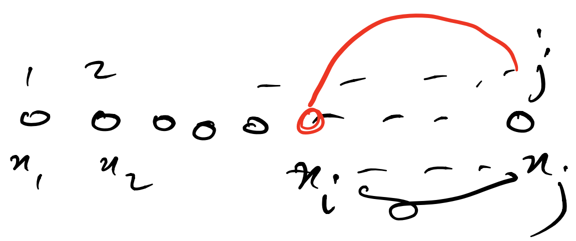
(b) $j_1 > j_2$.

Not allowed. $\times$

Goal: Given RNA strand of length n, find secondary structure of longest cardinality.

Dynamic Programming to solve above problem:

$OPT(j)$: be the optimal secondary structure using the strand $x_1 \cdots x_j$.



→ j does not form a secondary bond.

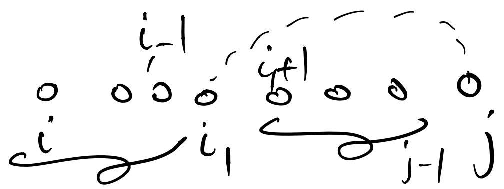
$$OPT(j) = OPT(j-1)$$

→ $$OPT(j) = OPT(i-1) + \underbrace{f(i+1, j-1)}_{??} + 1$$

Leads us to a more general subproblem space:

$OPT(i,j)$: be the optimal secondary structure using the strand $x_i \cdots x_j$.

$$\mathrm{OPT}(i,j) = \begin{cases} \mathrm{OPT}(i, j-1) & \text{(if } j \text{ does not participate)} \\ \text{if } \{x_{i_1}, x_j\} \text{ is 'valid'} \\ \mathrm{OPT}(i, i_1 - 1) + \mathrm{OPT}(i_1+1, j-1) + 1 \end{cases}$$



⊛

$$\mathrm{OPT}(i,j) = \max \left\{ \max_{i_1 \in \mathrm{good}(i,j)} \left\{ \mathrm{OPT}(i, i_1 - 1) + \mathrm{OPT}(i_1+1, j-1) + 1 \right\}, \; \mathrm{OPT}(i, j-1) \right\}$$

$$\mathrm{good}(i,j) = \begin{cases} [\, i, j-5 \,] & \text{if } i < j-4 \\ \emptyset & \text{otherwise} \end{cases}$$

Output $\mathrm{OPT}(1, n)$.

Pseudocode:

```
for k = 1 to n-1
    for i = 1 to n-k

        compute OPT(i, i+k) using (*)
    endfor
endfor
Output OPT(1, n)
```

$O(n^3)$ running time.

Solving recurrences.

easy: $f(0) = c$
$f(n) = f(n-1) + c'$.

Unrolling recursions:

$$f(n) = f(n-1) + c'$$
$$= f(n-2) + 2c'$$
$$\vdots$$
$$f(n) = f(0) + nc' = \Theta(n).$$

$\rightarrow$ $f(1) = c$

$f(n) = f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + c'$.

unroll $\log n$ times (levels)

$f(n) = c' \log n + f(1)$

$= O(\log n)$

$\rightarrow$ $T(n) = 2\,T(n-1) + 1$; $T(0) = 0$.

'Guess method'.

say : $T(n) = cn$ for some constant c .

$T(n) = 2\,T(n-1) + 1$

$\leq 2\,c(n-1) + 1$ (using induction)

$\not\leq cn$

say: $T(n) = 2^n - 1$

$$T(n) \leq 2(2^{n-1} - 1) + 1$$
$$= 2^n - 1 < 2^n \quad \checkmark .$$

$$T(n) = O(2^n).$$

$\rightarrow \quad T(n) = \sqrt{n}\, T(\sqrt{n}) + n.$

$$T(n) = n^{3/4}\, T(n^{1/4}) + 2n$$

$$T(n) = \sqrt{n}\left(n^{1/4}\, T(n^{1/4}) + \sqrt{n}\right) + n$$

$$T(n) = O(n \log\log n).$$