14 February 2024. RNA Folding

Plan.

* RNA Folding Problem

* Announcements

* A different flavor of DP.

Another problem from biology

<u>RNA Sequences</u> $m \in \{A, U, C, G\}^*$

$m = A\,U\,U\,C\,G\,G\,A\,C\,G\,G\,A\,A$

<u>Key Question</u>:

What <u>structure</u> does the RNA molecule adopt?

What structure does the RNA molecule adopt?

↳ Base pairing

A — U

C — G

$$BP = \{(A,U), (U,A), (C,G), (G,C)\}$$

What structure does the RNA molecule adopt?

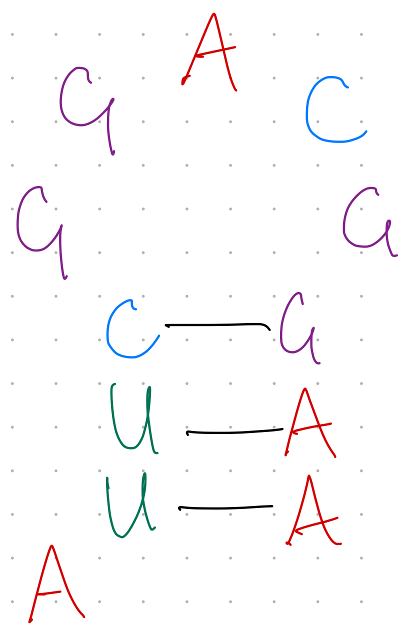
m = A U U C G G A C G G A A

$\downarrow$ base pairing

What structure does the RNA molecule adopt?

m = A U U C G G A C G G A A

$\downarrow$ base pairing



<u>Given</u>: RNA Sequence $m \in \{A, U, C, G\}^*$

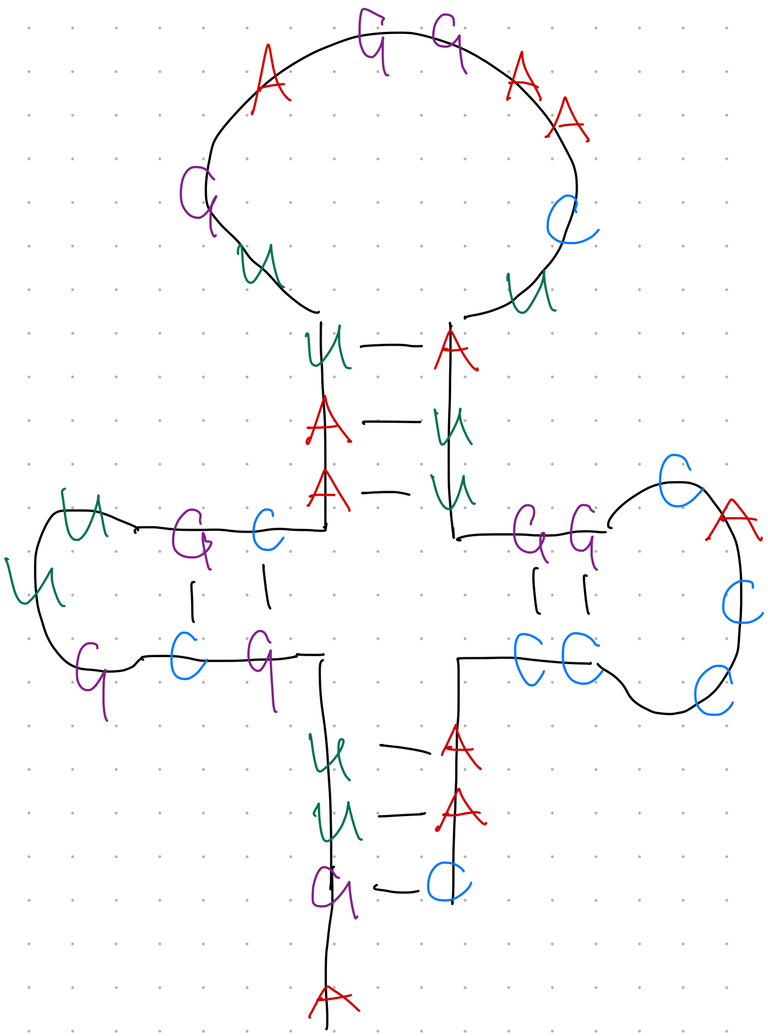
<u>Find</u>: Minimum "energy" structure

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

Given: RNA sequence $m \in \{A, U, C, G\}^*$

Find: Minimum "energy" structure

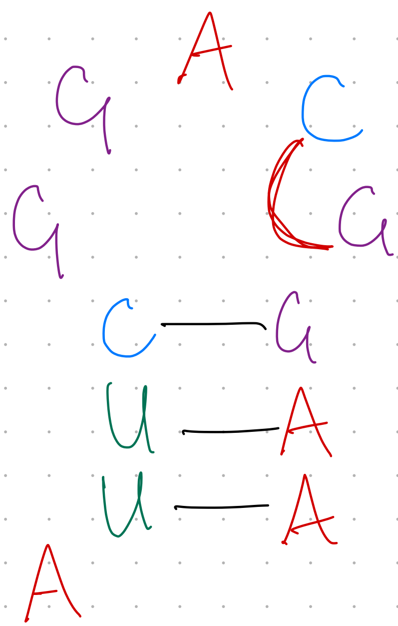
AGUUGCGUUGCAAUUGAGGAACUAUUGGCACCCCAAC



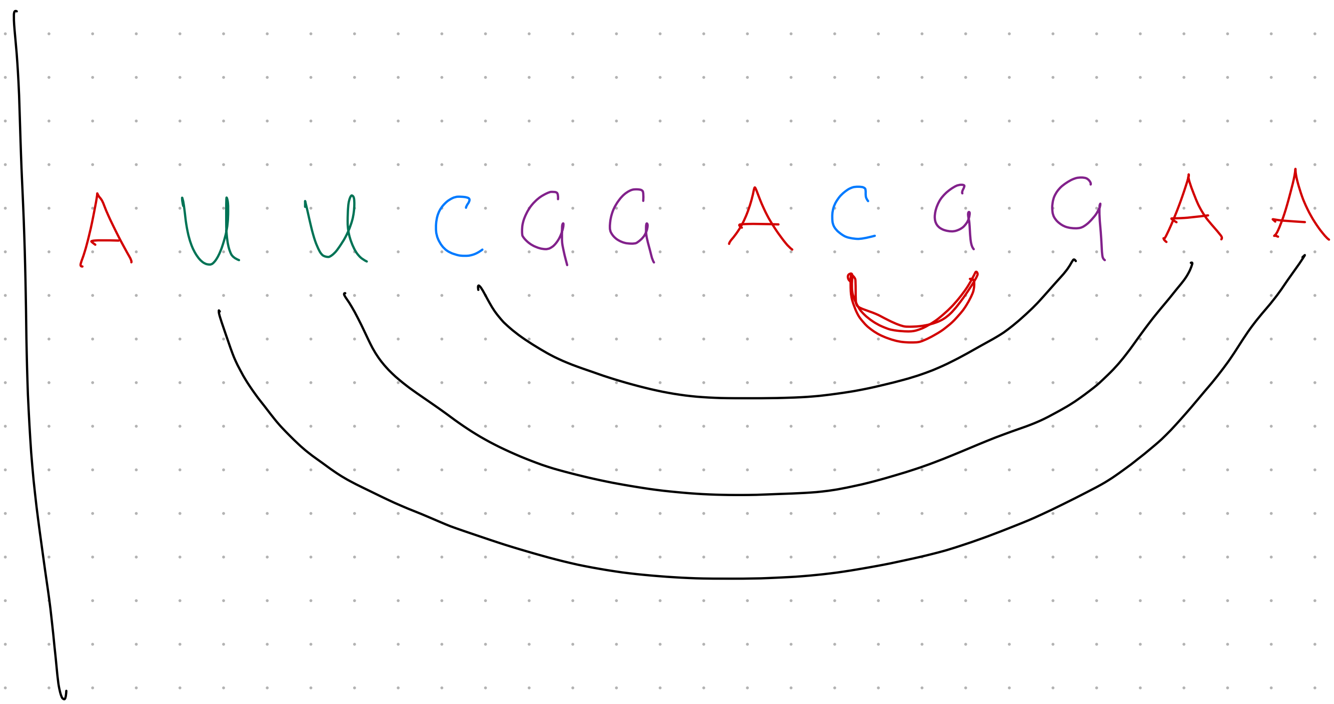
Min "energy" $\equiv$ Max "legal" base pairing

Given: RNA sequence $m \in \{A, U, C, G\}^*$

Compute: Maximum Non-crossing matching E of preferred base pairs within m.



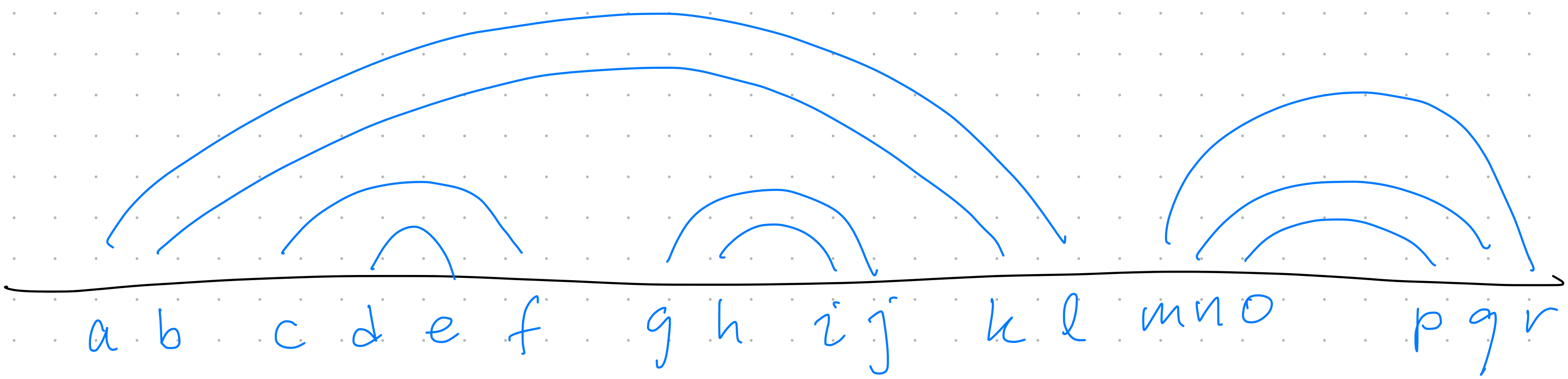
$\Rightarrow$



Given: RNA sequence $m \in \{A, U, C, G\}^*$

Compute: Maximum Non-crossing matching E of preferred base pairs within m.

$m =$



$$E = \{ (a,l), (b,k), (c,f), (d,e), \ldots \quad (m,r), (n,q), (o,p) \}$$

<u>Given</u>: RNA sequence $m \in \{A, U, C, G\}^*$

<u>Compute</u>: Maximum <u>Non-crossing</u> matching E of preferred base pairs within m.

<u>Non-crossing</u>: if $(i, j), (k, \ell) \in E$

then, $\neg (i < k < j < \ell)$.

$m_i \cdots m_k \cdots m_j \cdots m_\ell$

Announcements

* HW1 Grades Released
* HW3 Due Tomorrow 11:59pm
* Prelim Next Week

  Feb 20, 7:30 pm

  Uris G01, Olin 155, Olin 165

  See Ed for room assignment

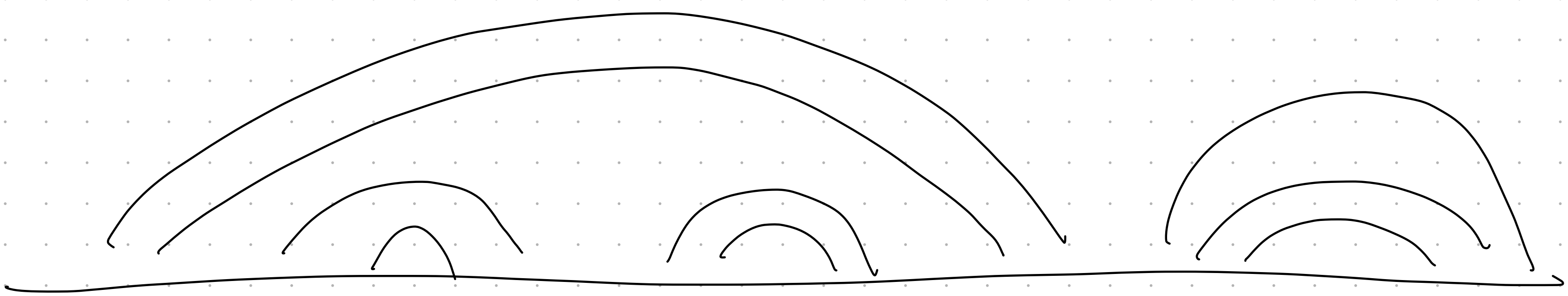
* Prelim Reviews: Gates G01

  Saturday 1-3:30 pm

  Sunday 2-4:30 pm

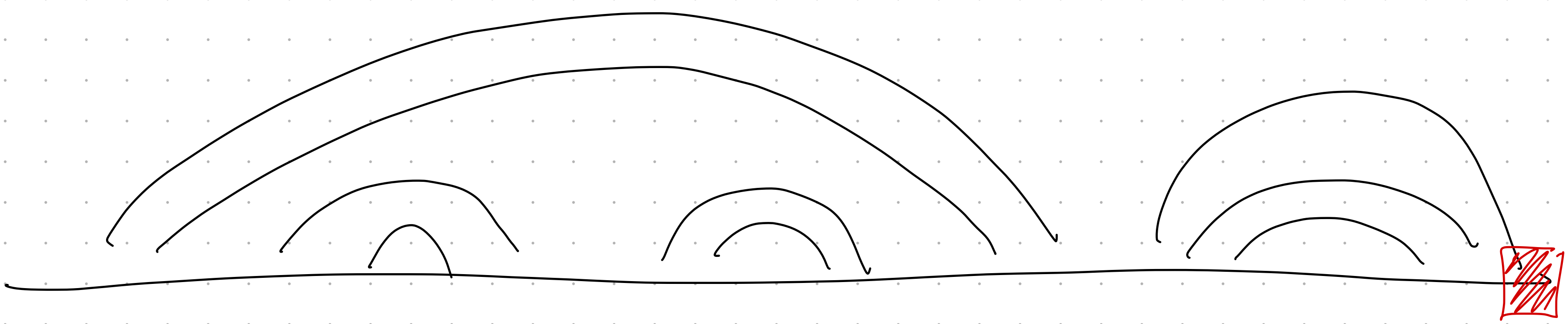
Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions

$m =$ 

Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions

$m =$ 

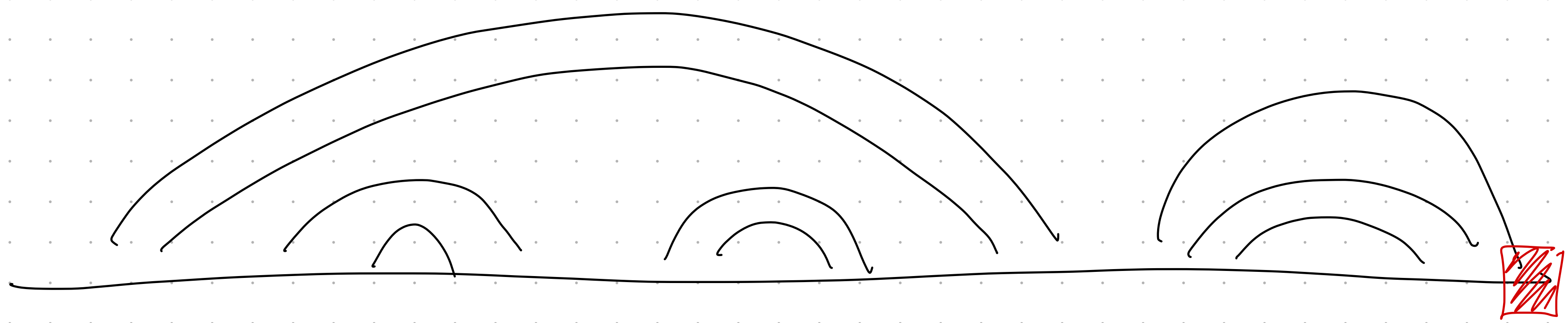
$m_n \in E$

OR

$m_n \notin E$

Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions

$m =$ 

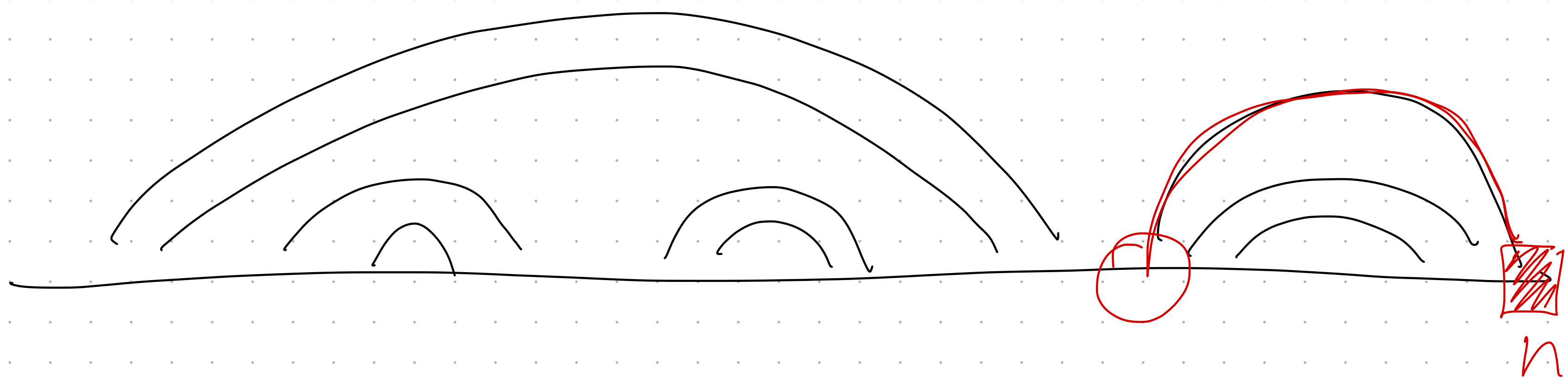
$m_n \in E$

OR

$OPT[n] = OPT[n-1] \longleftarrow m_n \notin E$

Dynamic Programming Approach

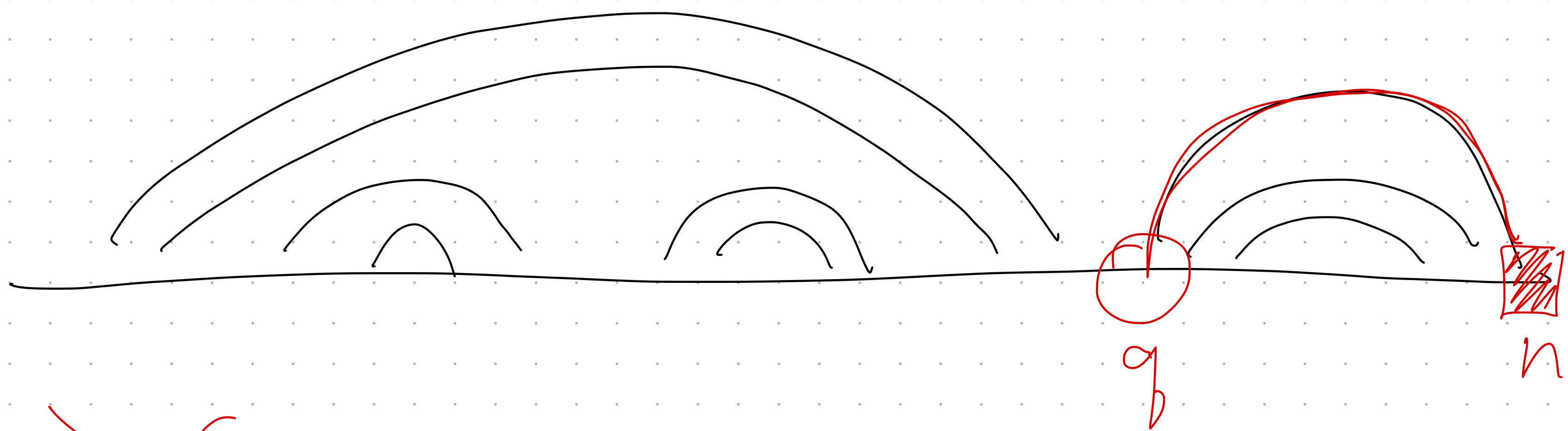
* Express Complete Solution in terms of subsolutions

$m =$ 

$m_n \in E$

Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions

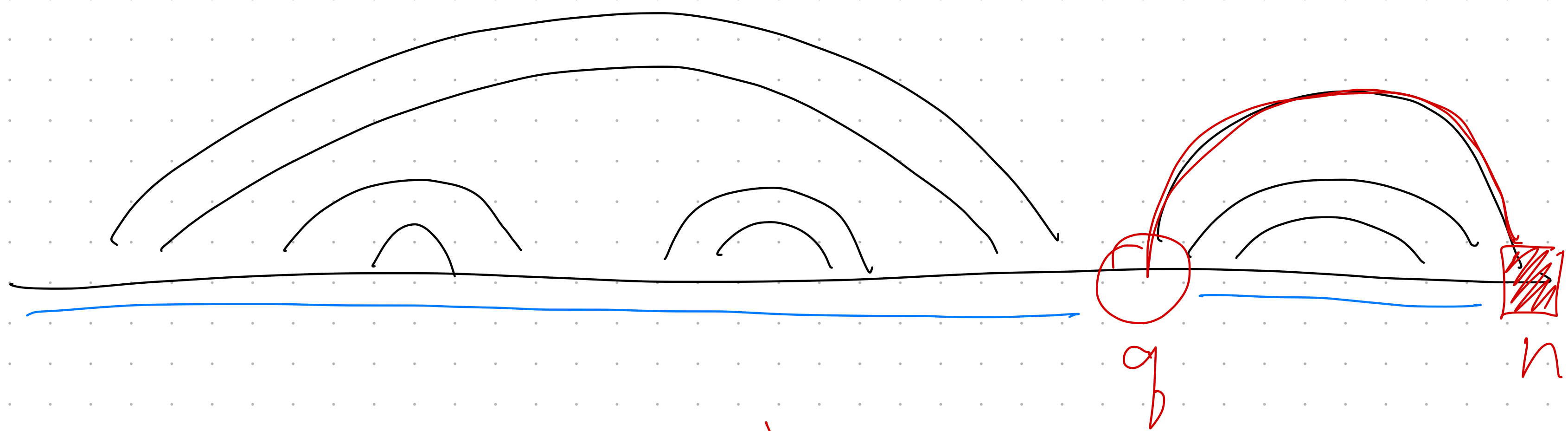
$m =$ 

$\cancel{m_n \in E}$

$\exists\, q \in [n]$ s.t. $(m_q, m_n) \in E$

Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions

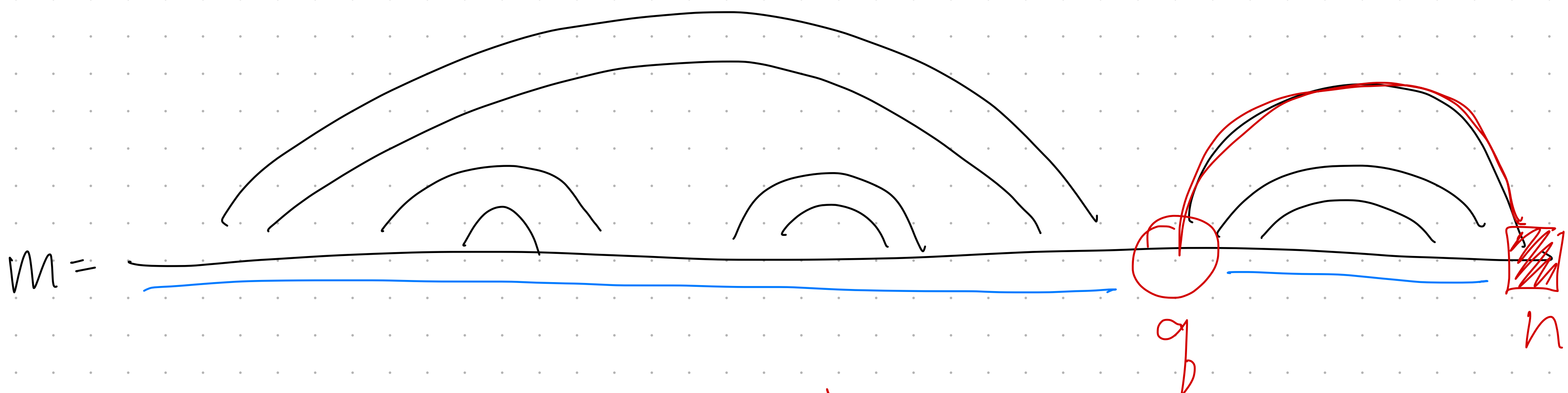
$m =$ 

$\exists q \in [n]$ s.t. $(m_q, m_n) \in E$

What are the subproblems?

Dynamic Programming Approach

* Express Complete Solution in terms of subsolutions



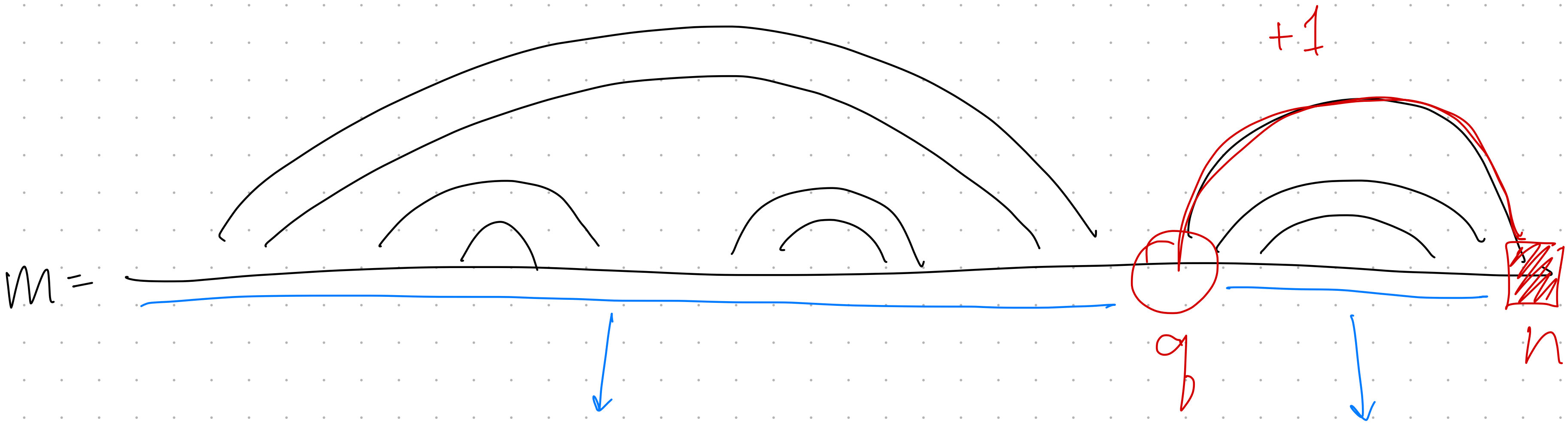
$\exists\, q \in [n]$ s.t. $(m_q, m_n) \in E$

$$OPT[n] = OPT[q-1] + \underline{\quad ?? \quad}$$

Solution: 2D Dynamic Programming Table

(even though "1D" problem)

$OPT[p, r]$ = Best non-crossing match on substring from $p \rightarrow r$

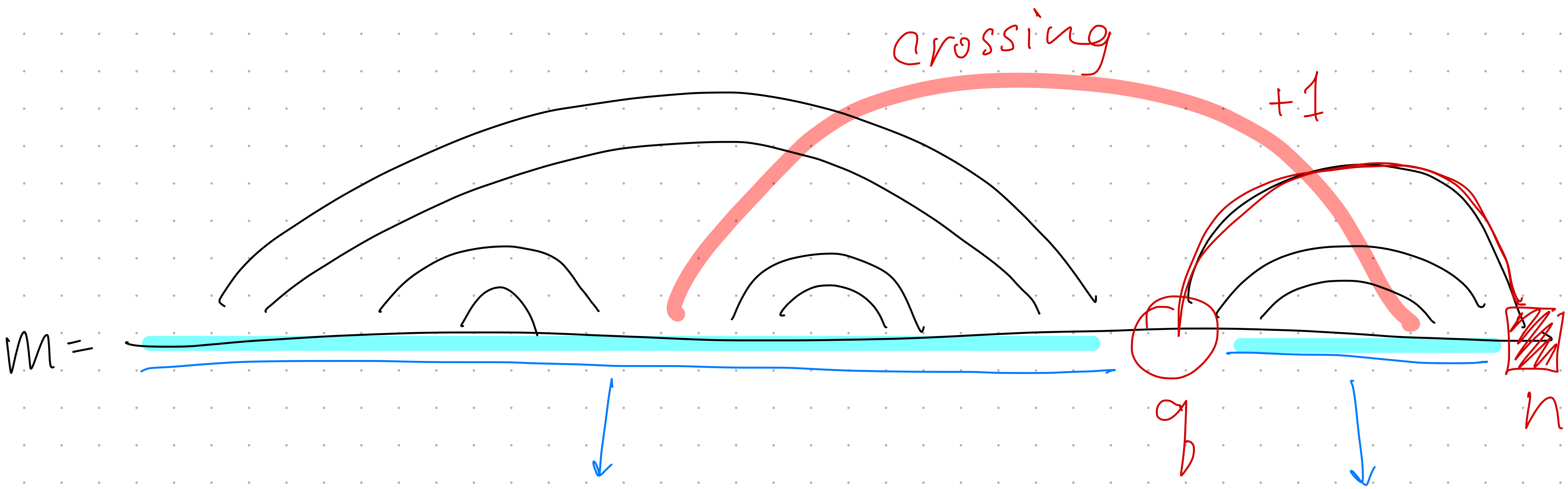


$OPT[1, q-1] \quad + \quad OPT[q+1, n-1]$

Solution 2D Dynamic Programming Table

(even though "1D" problem)

$OPT[p, r]$ = Best non-crossing match on substring from $p \to r$



$OPT[1, q-1] \quad + \quad OPT[q+1, n-1]$

DP Recurrence: Fix $p < r$.

$OPT[p, r] =$

① M_r not involved

$OPT[p, r-1]$

② M_r involved

$\Rightarrow$ Find best legal q

DP Recurrence. Fix $p < r$.

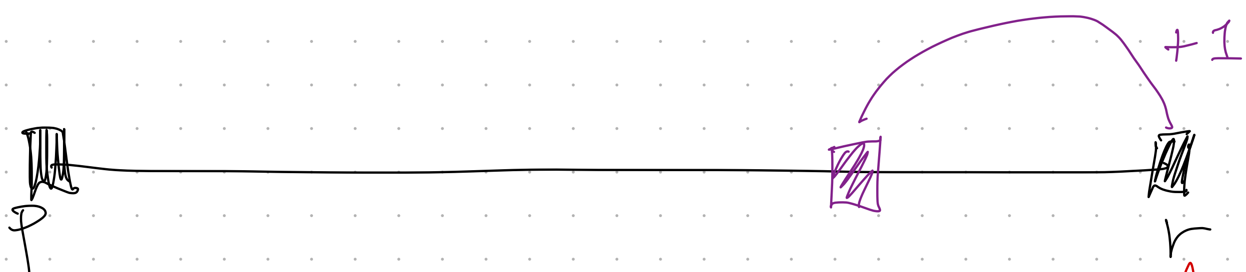
$OPT[p, r] =$

① M_r not involved

$OPT[p, r-1]$

② M_r involved

$\Rightarrow$ Find best legal q



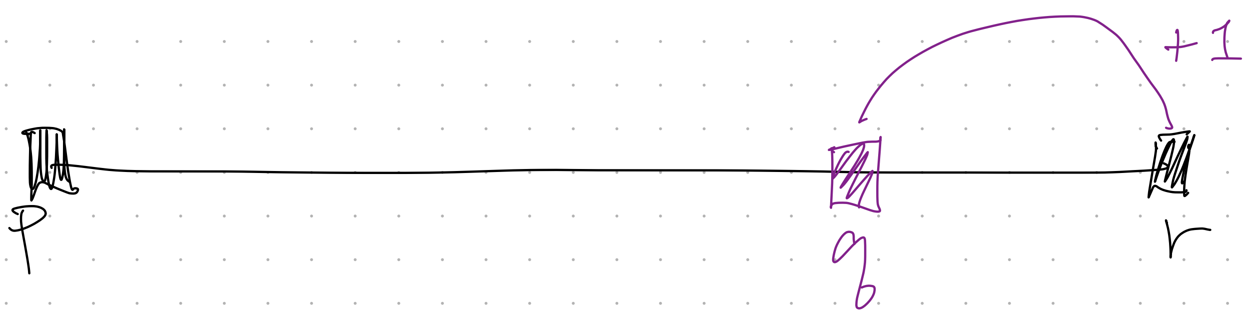
$$BP = \left\{ \begin{array}{ccc} U & — & A \\ A & — & U \\ G & — & C \\ C & — & G \end{array} \right\}$$

DP Recurrence. Fix $p < r$.

① M_r not involved
$OPT[p, r-1]$

$OPT[p, r] =$

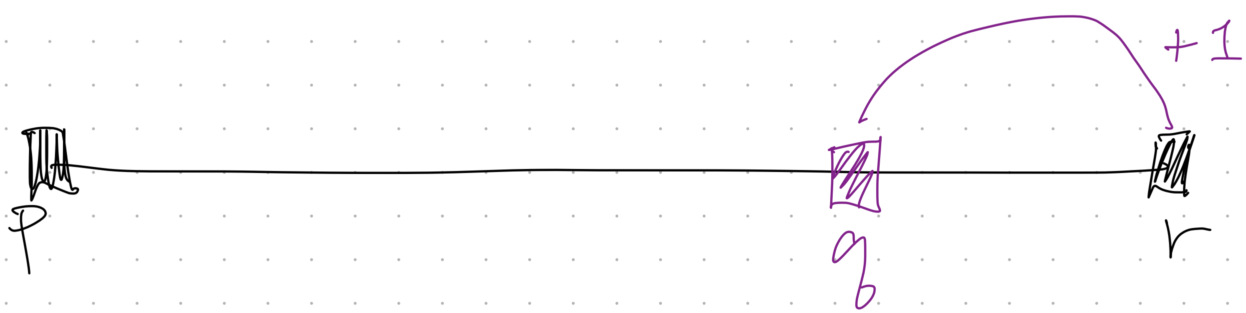
② M_r involved
$\Rightarrow$ Find <u>best</u> legal q



$$\max_{\substack{q \in [p+1, r-1] \\ \text{s.t. } (M_q, M_r) \in BP}} \left\{ 1 + OPT[p, q-1] + OPT[q+1, r-1] \right\}$$

DP Recurrence. Fix $p < r$.

$$\mathrm{OPT}[p, r] = \begin{cases} \text{① } M_r \text{ not involved:} & \mathrm{OPT}[p, r-1] \\ \text{② } M_r \text{ involved} \implies \text{Find } \underline{\text{best}} \text{ legal } q & \end{cases}$$



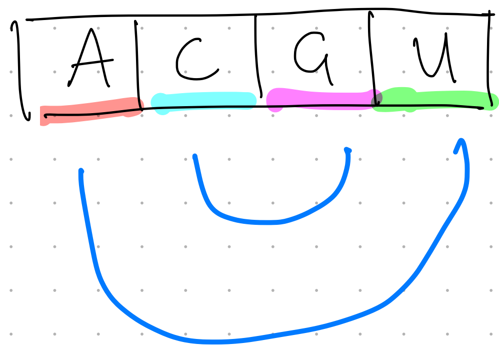
$$\max_{\substack{q \in [p+1, r-1] \\ \text{s.t. } (M_q, M_r) \in BP}} \left\{ 1 + \mathrm{OPT}[p, q-1] + \underbrace{\mathrm{OPT}[q+1, r-1]}_{\text{edge cases?}} \right\}$$

DP Recurrence.

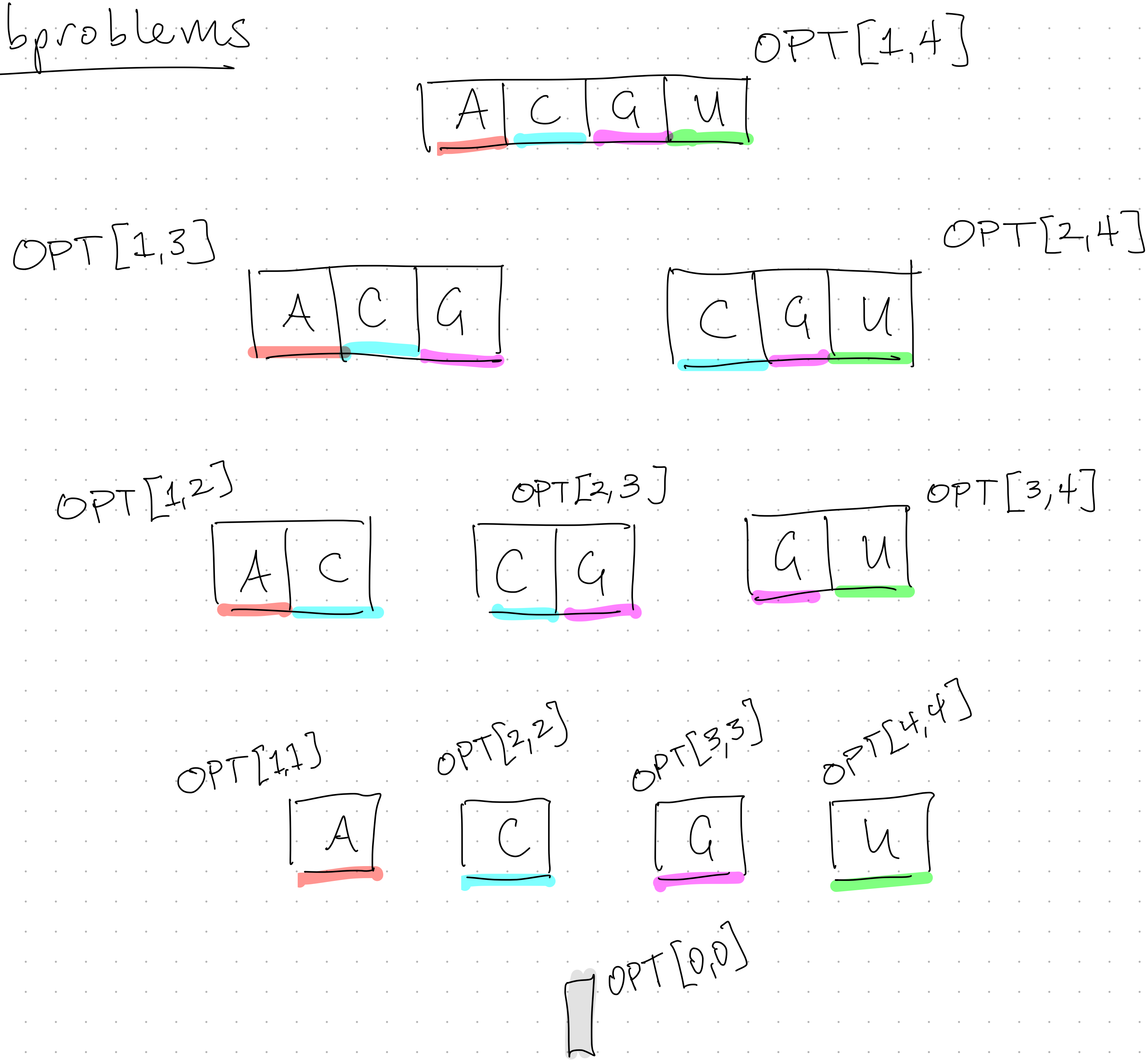
$$OPT[p, r] = 0 \quad \text{if } p \geq r$$

$$OPT[p, r] = \max \begin{cases} OPT[p, r-1], \\ \max\limits_{\substack{q \in [p, r-1] \\ (M_q, M_r) \in B}} \{ 1 + OPT[p, q-1] + OPT[q+1, r] \} \end{cases} \quad \text{if } p < r.$$

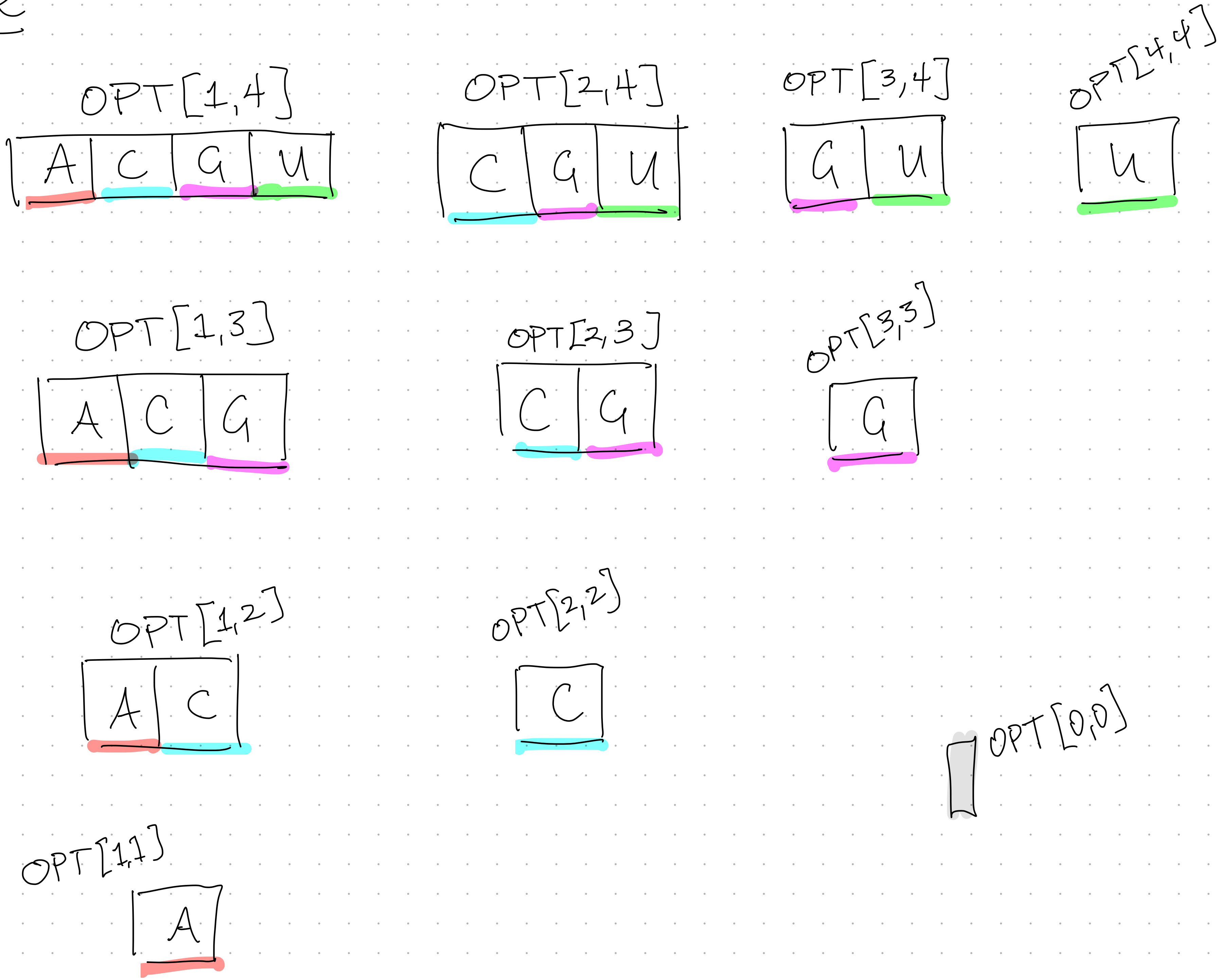
Subproblems



Subproblems



Table


OPT[1,4]
A C G U
OPT[2,4]
C G U
OPT[3,4]
G U
OPT[4,4]
U
OPT[1,3]
A C G
OPT[2,3]
C G
OPT[3,3]
G
OPT[1,2]
A C
OPT[2,2]
C
OPT[0,0]
OPT[1,1]
A

Table

OPT[1,4] A C G U	OPT[2,4] C G U	OPT[3,4] G U	OPT[4,4] U
OPT[1,3] A C G	OPT[2,3] C G	OPT[3,3] G	
OPT[1,2] A C	OPT[2,2] C		OPT[0,0]
OPT[1,1] A			

Proof of Correctness. By induction on the width of interval considered in OPT.

Base Case. $w=0$, $w=1$.

$OPT[p,p]=0$ $\quad$ $OPT[p,p+1]=0$

// There are zero possible matchings between 0 and 1 characters. ✓

Proof of Correctness. By induction on the width of interval considered in OPT.

Base Case. $w=0$, $w=1$.

$OPT[p,p]=0$ $\quad$ $OPT[p,p+1]=0$

// There are zero possible matchings between 0 and 1 characters. ✓

Inductive Step

IH: $\forall$ widths $w < w_0$. $\forall r \in [w+1, n]$

$OPT[r-w, r] =$ Max feasible non-crossing base pairs from $[r-w, r]$

Proof of Correctness. By induction on the width of interval considered in OPT.

Base Case. $w=0$, $w=1$.

$\downarrow$ $OPT[p,p]=0$ $\quad$ $\downarrow$ $OPT[p,p+1]=0$

// There are zero possible matchings between 0 and 1 characters. ✓

Inductive Step IH: $\forall w < w_0 \;\; \forall r \in [w+1, n]$, $OPT[r-w, r]$ = Max non-crossing base pairs.

We show our recurrence considers all feasible base pairings and takes the maximum.

Proof of Correctness. By induction on the width of interval considered in OPT.

Base Case. $w=0$, $w=1$.

$OPT[p,p]=0$ $OPT[p,p+1]=0$

// There are zero possible matchings between 0 and 1 characters. ✓

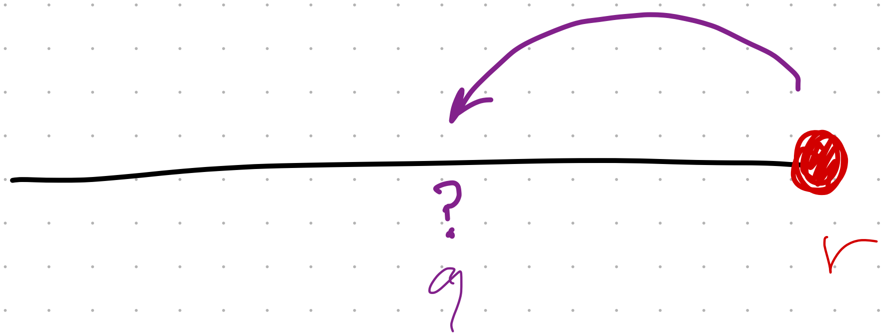
Inductive Step IH: $\forall w < w_0$ $\forall r \in [w+1, n]$, $OPT[r-w, r]$ = Max non-crossing base pairs.

We show our recurrence considers all feasible base pairings, and takes the maximum.

Case 1: M_r NOT involved $\Rightarrow$ $OPT[r-w, r-1]$ is optimal (already computed)

Case M_r is involved.

Recurrence considers all feasible matches in BP for M_r



Suppose we find and match M_r to some M_q.

Then, by non-crossing

No base pairs are allowed with endpoints

$$(p, p') \text{ s.t. } \quad p < q < p' < r.$$

$\Rightarrow$ all remaining feasible solns. have matches between

$$[1 \to q-1] \text{ and } [q+1 \to r-1].$$

Our Recurrence Reflects this.

RNA-Fold(m)

For $p = 1 \to n$: $OPT[p, p] = 0$.

For width $= 1 \to n-1$ // for each width

For $p = 1 \to n -$ width // go through each window $[p, r]$ of the width

$r = p +$ width

RNA-Fold (n)

For $p = 1 \to n$: $OPT[p, p] = 0$.

For width $= 1 \to n-1$

  For $p = 1 \to n - \text{width}$

    $r = p + \text{width}$

    prev $= OPT[p, r-1]$

    match $= 0$

    → pick best match

$$OPT[p, r] = \max \{ \text{prev}, \text{match} \}$$

RNA-Fold(m)

```
For p = 1 → n : OPT[p, p] = 0.

For width = 1 → n-1
    For p = 1 → n - width
        r = p + width
        prev = OPT[p, r-1]
        match = 0
        For q = p → r-1
        |   if (M_q, M_r) ∈ BP :
        |       match = max { match, 1 + OPT[p, q-1]
        |                            + OPT[q+1, r] }
        OPT[p, r] = max { prev, match }
```

RNA-Fold(m)

For $p = 1 \to n$: $OPT[p,p] = 0$.

Running Time?

For width $= 1 \to n-1$

  For $p = 1 \to n - \text{width}$

    $r = p + \text{width}$

    prev $= OPT[p, r-1]$

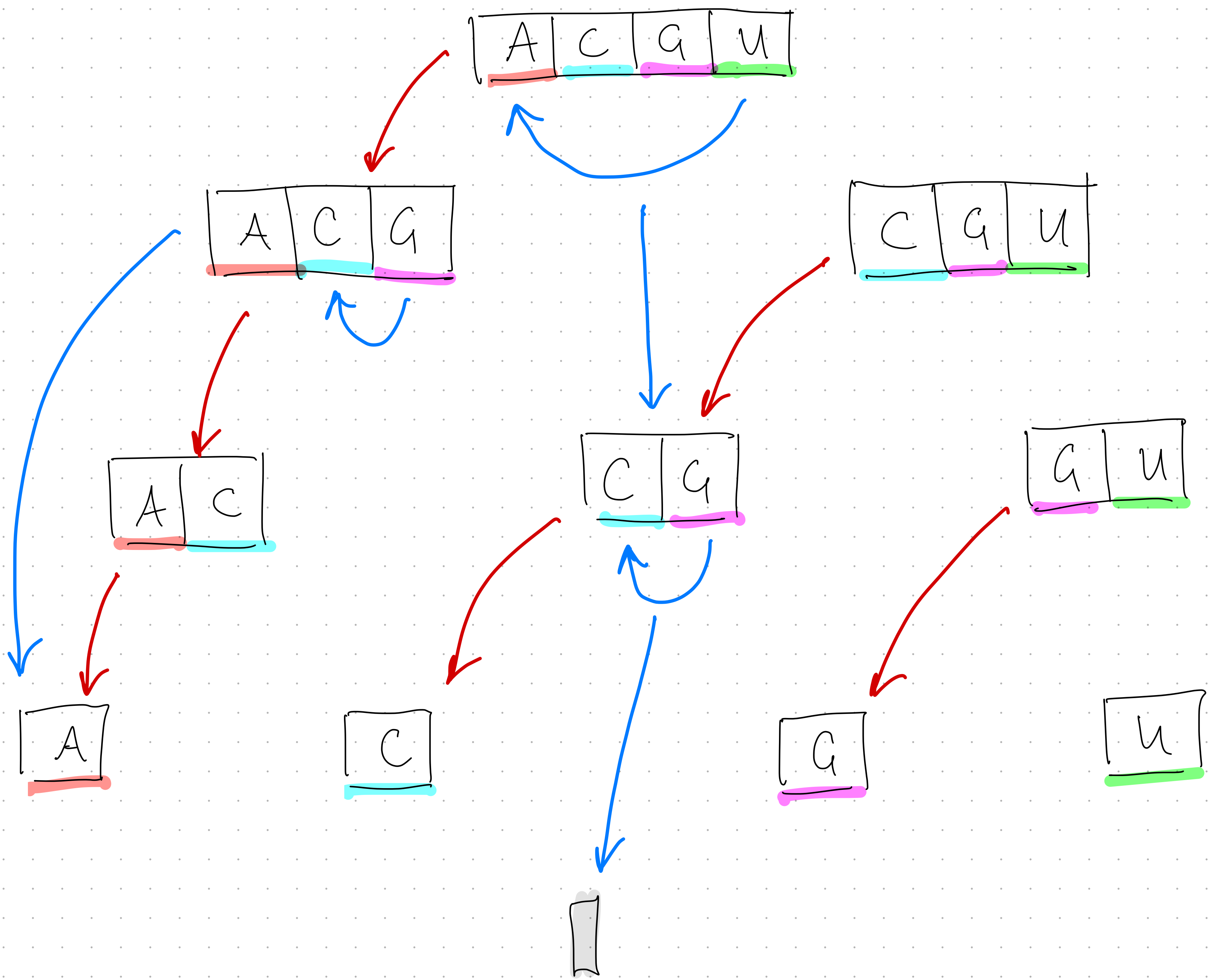
    match $= 0$

    For $q = p \to r-1$

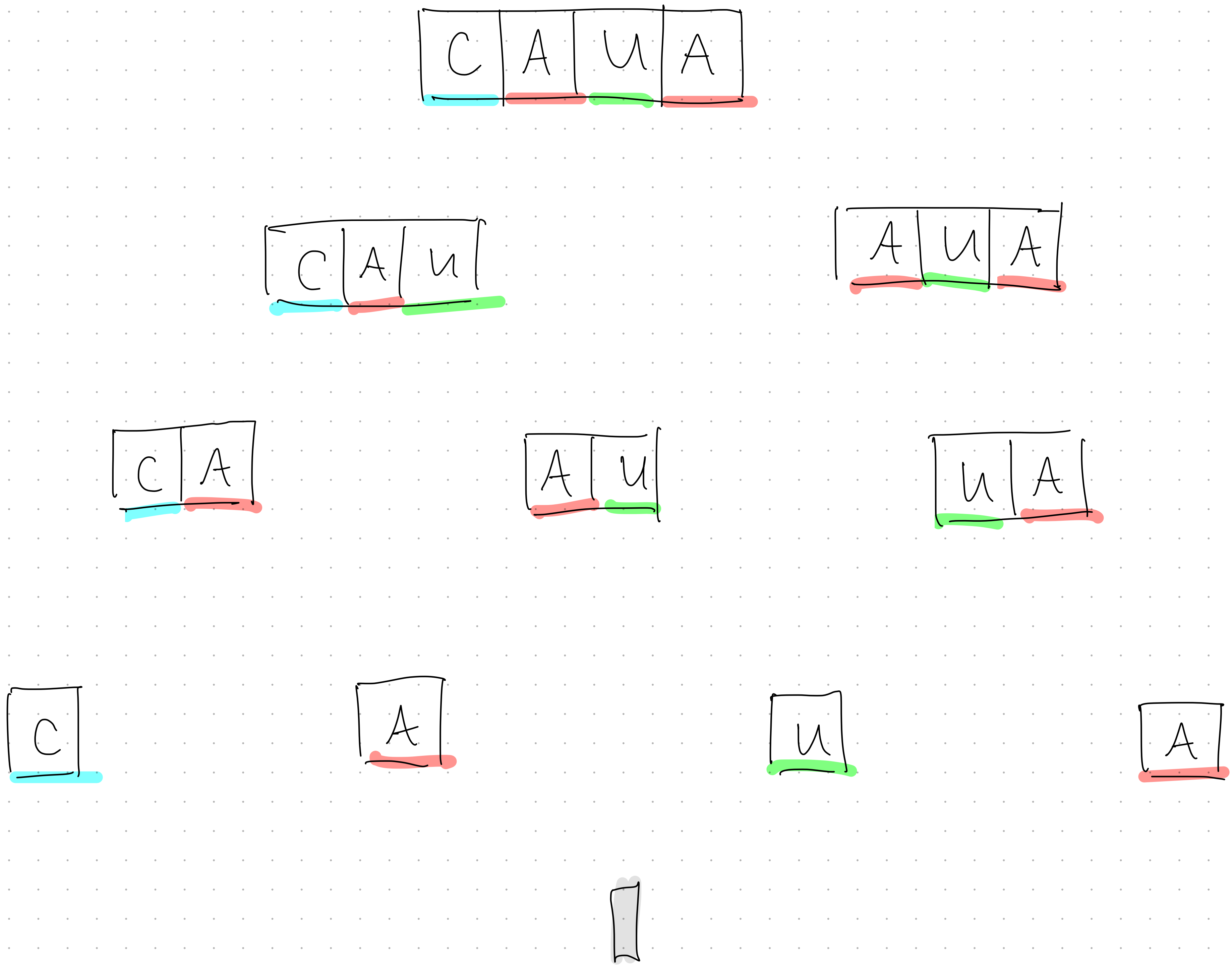
      if $(M_q, M_r) \in BP$:

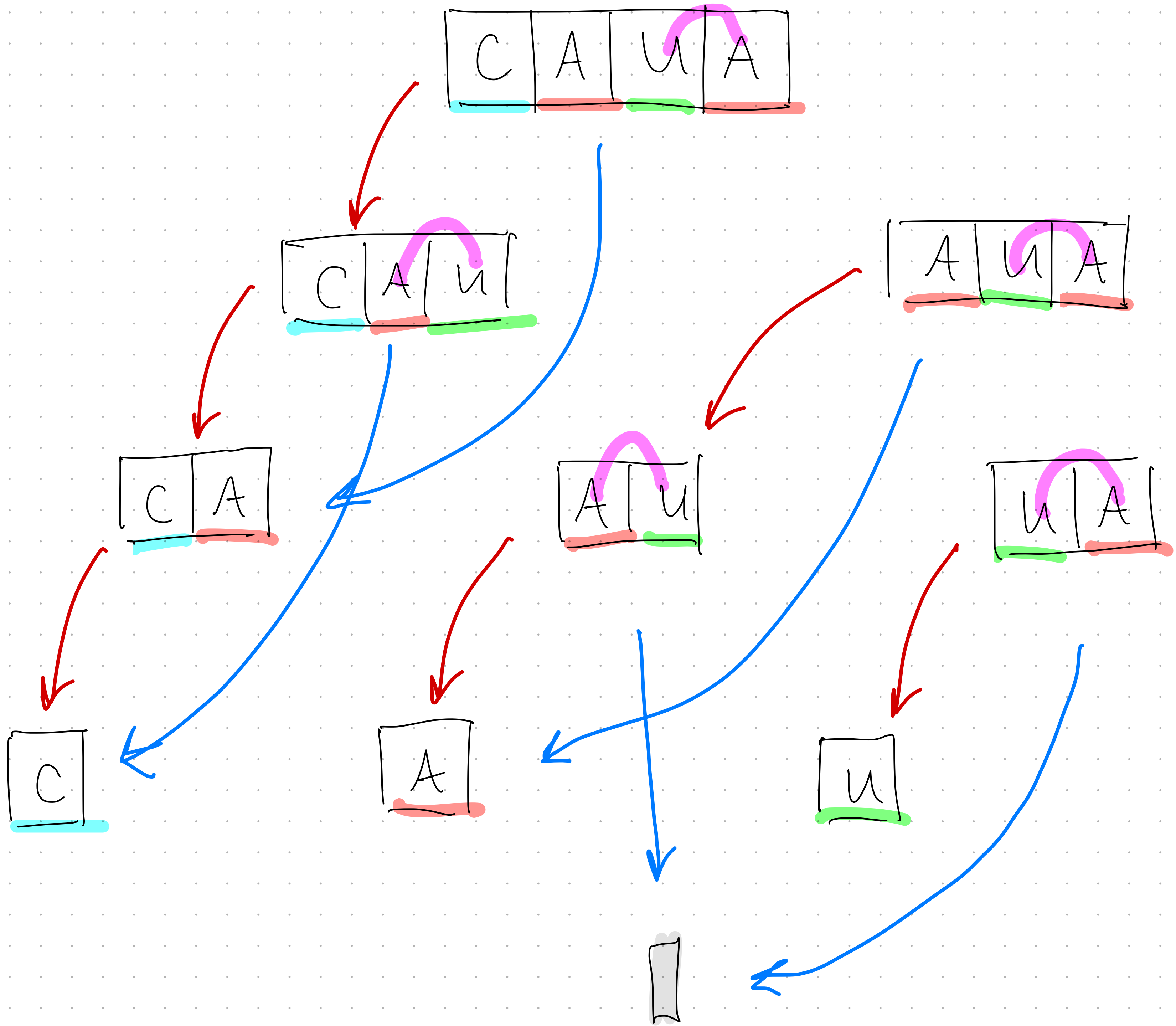
        match $= \max \left\{ \text{match}, \; \begin{array}{l} 1 + OPT[p, q-1] \\ + OPT[q+1, r] \end{array} \right\}$

    $OPT[p, r] = \max \{ \text{prev}, \text{match} \}$

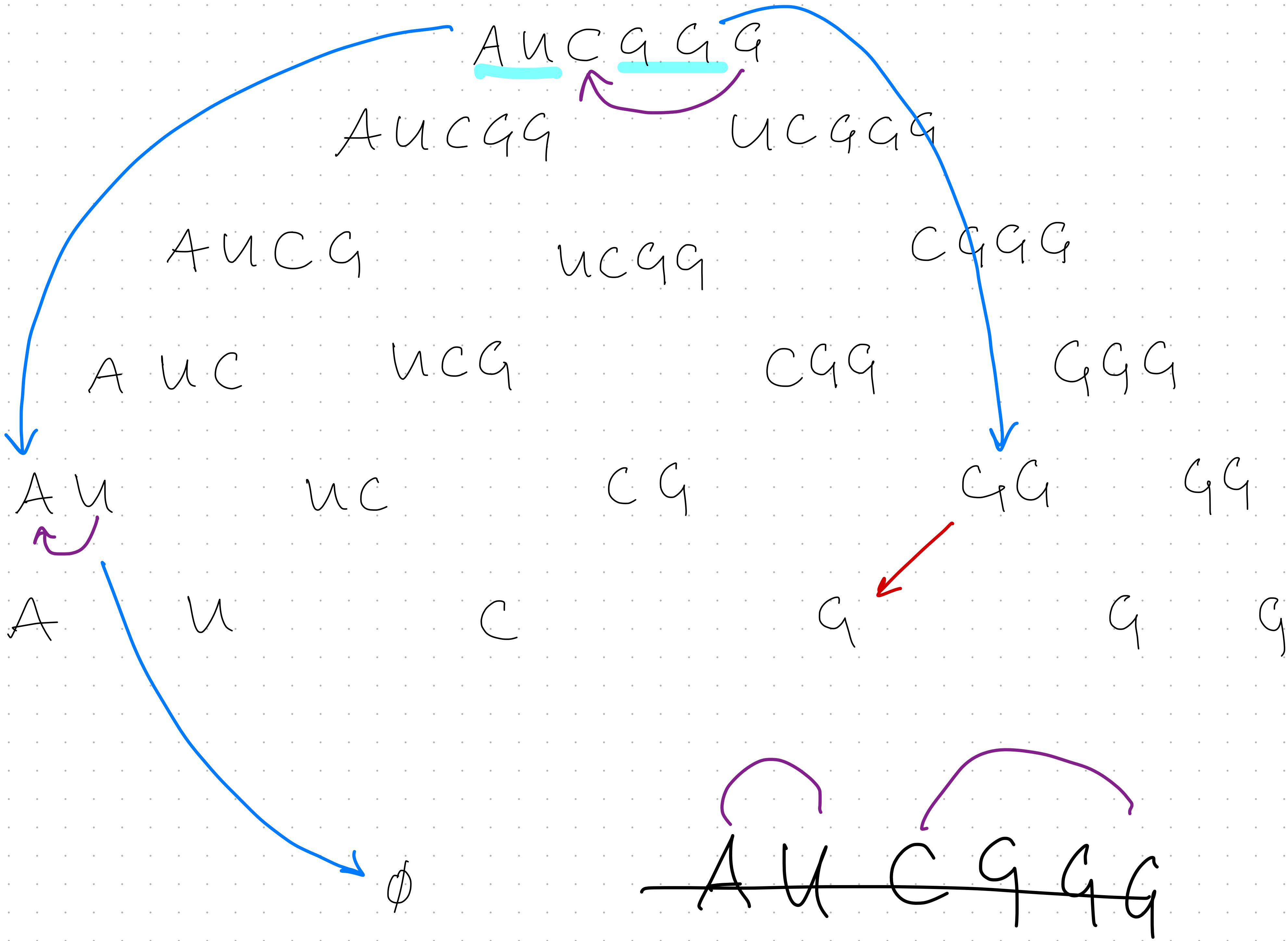


A C G U
A C G
C G U
A C
C G
G U
A
C
G
U


C A U A
C A U
A U A
C A
A U
U A
C
A
U
A



C A U A
C A U
A U A
C A
A U
U A
C
A
U
A


AUCGGG
AUCGG
UCGGG
AUCG
UCGG
CGGG
AUC
UCG
CGG
GGG
AU
UC
CG
GG
GG
A
U
C
G
G
G
∅
AUCGGG

$OPT[1,4]$

$OPT[1,3]$ $OPT[2,4]$

$OPT[1,2]$ $OPT[2,3]$ $OPT[3,4]$

$OPT[1,1]$ $OPT[2,2]$ $OPT[3,3]$ $OPT[4,4]$

$\emptyset$

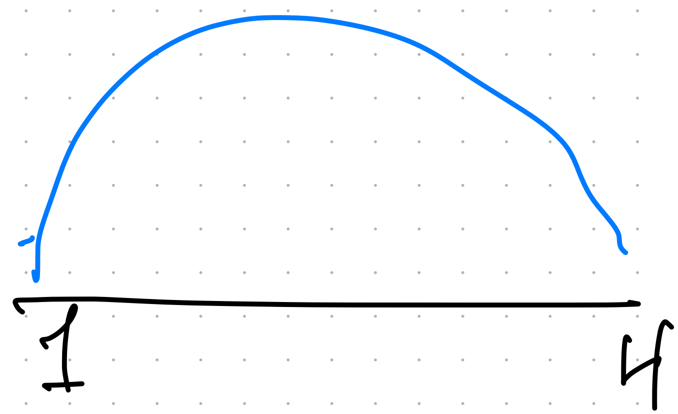
$OPT[1,4]$

$OPT[1,3]$ $\quad$ $OPT[2,4]$

$OPT[1,2]$ $\quad$ $OPT[2,3]$ $\quad$ $OPT[3,4]$

$OPT[1,1]$ $\quad$ $OPT[2,2]$ $\quad$ $OPT[3,3]$ $\quad$ $OPT[4,4]$

$\emptyset$



RNA-Fold(m)

$$\underbrace{\text{BACK}[p,r]}_{\text{which subproblems yield } OPT[p,r]}$$

For $p = 1 \to n$: $OPT[p,p] = 0$.

For width $= 1 \to n-1$

  For $p = 1 \to n - \text{width}$

    $r = p + \text{width}$

    prev $= OPT[p, r-1]$

    match $= 0$

    For $q = p \to r-1$

      if $(M_q, M_r) \in BP$:

        $$\text{match} = \max \left\{ \text{match},\ 1 + OPT[p, q-1] + OPT[q+1, r] \right\}$$

    $OPT[p,r] = \max\{\text{prev}, \text{match}\}$

RNA-Fold(M)

For $p = 1 \to n$: $OPT[p,p] = 0$.

For width $= 1 \to n-1$

  For $p = 1 \to n - \text{width}$

    $r = p + \text{width}$

    prev $= OPT[p, r-1]$

    match $= 0$

    For $q = p \to r-1$

      if $(M_q, M_r) \in BP$:

        match $= \max \left\{ \text{match}, \; 1 + OPT[p, q-1] + OPT[q+1, r] \right\}$

    $OPT[p,r] = \max \{ \text{prev}, \text{match} \}$

$\underbrace{BACK[p,r]}_{\text{which subproblems yield } OPT[p,r]}$

$BACK[p,r] = r-1$

RNA-Fold(m)

For $p = 1 \to n$: $OPT[p, p] = 0$.

For $width = 1 \to n-1$

  For $p = 1 \to n - width$

    $r = p + width$

    $prev = OPT[p, r-1]$

    $match = 0$

    For $q = p \to r-1$

      if $(M_q, M_r) \in BP$.

        $match = \max \left\{ match, \; 1 + OPT[p, q-1] + OPT[q+1, r] \right\}$

    $OPT[p, r] = \max \{ prev, match \}$

$\underbrace{BACK[p, r]}_{\text{which subproblems yield } OPT[p,r]}$

$BACK[p, r] = r-1$

$BACK[p, r] = q$

from last update