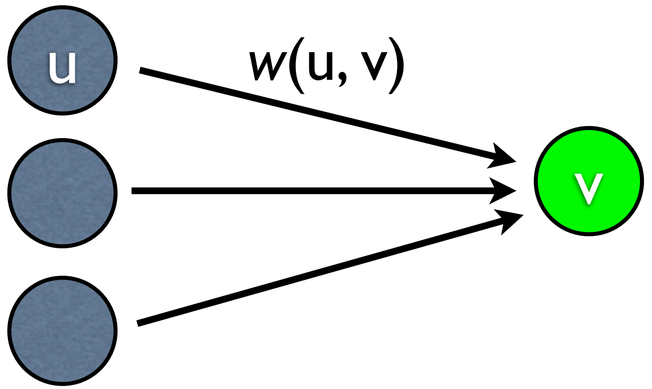
Viterbi Algorithm for DAGs

1. topological sort
2. visit each vertex v in sorted order and do updates
   - for each incoming edge (u, v) in E
   - use d(u) to update d(v): $d(v) \oplus = d(u) \otimes w(u, v)$
   - key observation: d(u) is fixed to optimal at this time

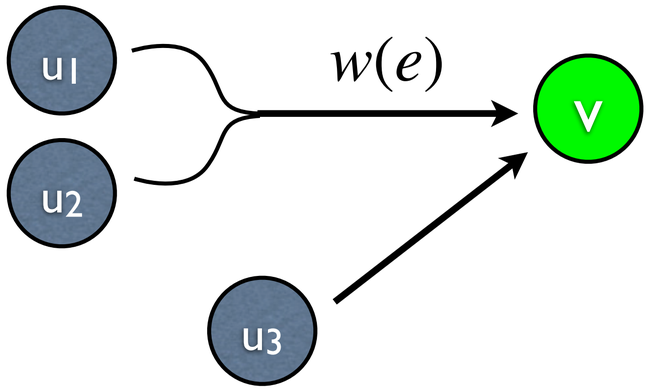


$d(v) \oplus = d(u) \otimes w(u, v)$

- time complexity: O(V + E)

Generalized Viterbi for DAHs (Hypergraphs)

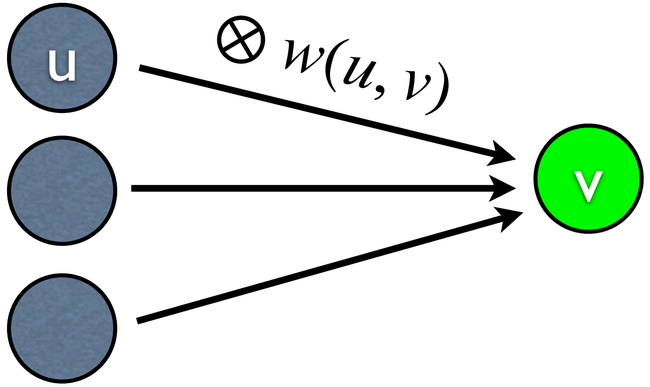
1. topological sort
2. visit each vertex v in sorted order and do updates
   - for each incoming hyperedge e = ((u1, .., u|e|), v, w(e))
   - use d(ui)'s to update d(v)
   - key observation: d(ui)'s are fixed to optimal at this time



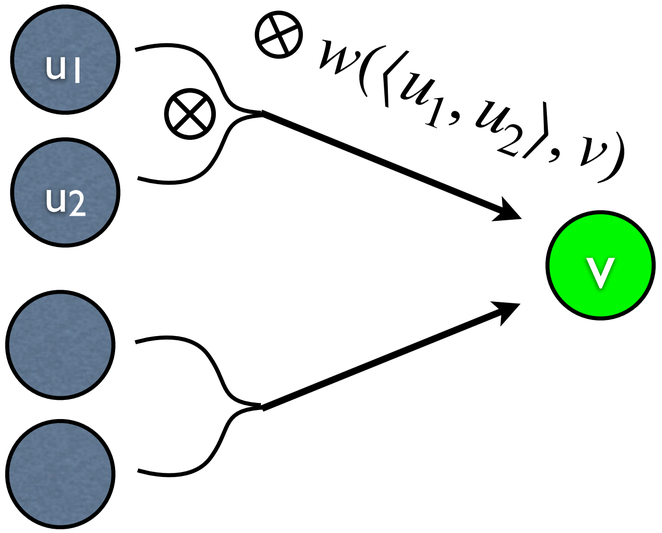
$$d(v) \oplus = d(u_1) \otimes d(u_2) \otimes w(e)$$

- time complexity: O(V + E) (assuming constant arity)

Graphs vs. Hypergraphs

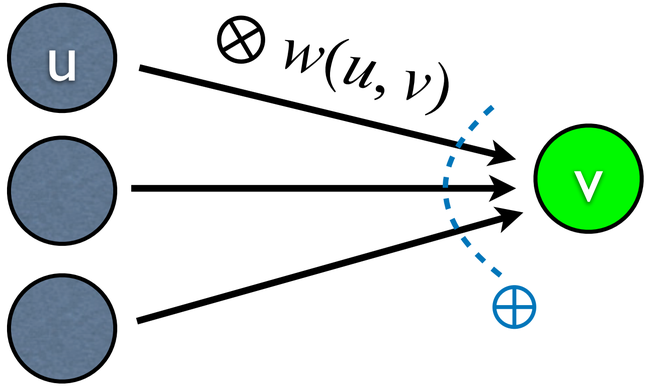


$$d(v) = \oplus_{(u,v)\in E}\, [d(u) \otimes w(u, v)]$$

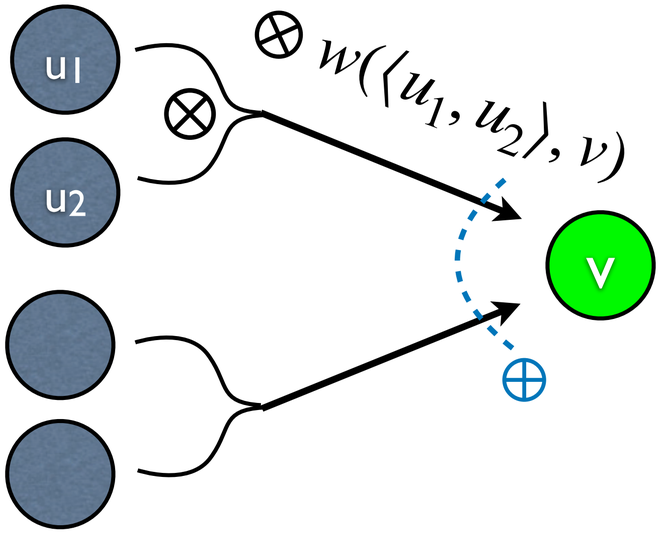


$$d(v) = \oplus_{(\langle u_1,u_2\rangle,v)\in E}\, [d(u_1) \otimes d(u_2) \otimes w(\langle u_1, u_2\rangle, v)]$$

Graphs vs. Hypergraphs



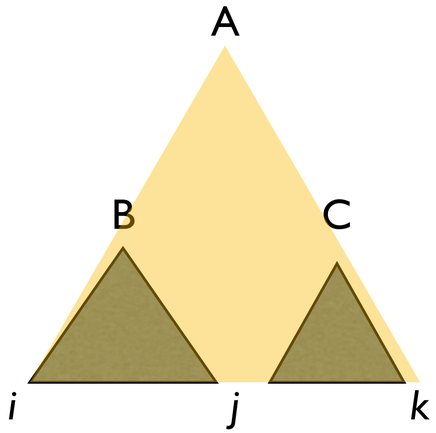
$$d(v) = \oplus_{(u,v)\in E} [d(u) \otimes w(u, v)]$$

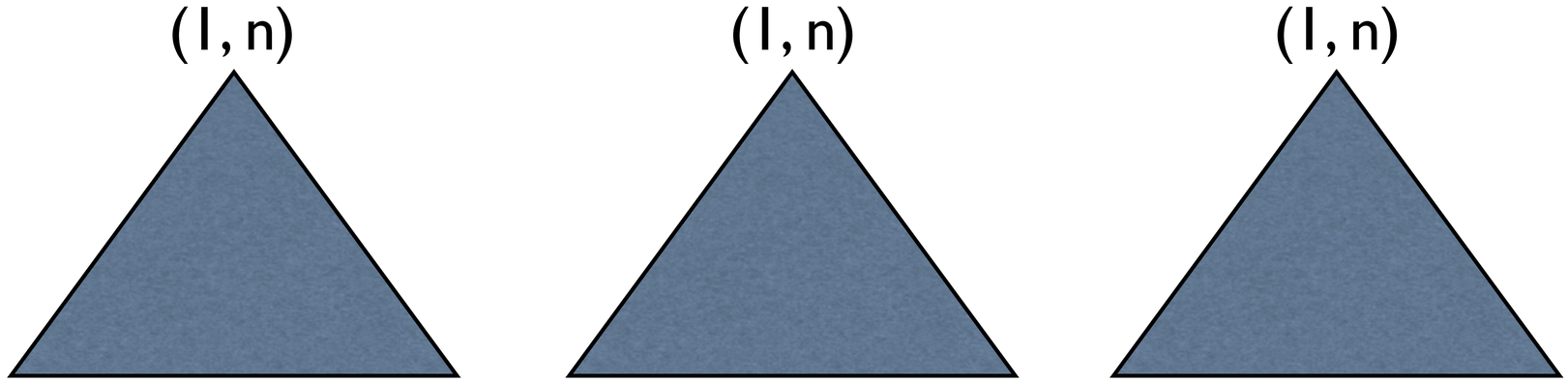


$$d(v) = \oplus_{(\langle u_1,u_2\rangle,v)\in E} [d(u_1) \otimes d(u_2) \otimes w(\langle u_1, u_2\rangle, v)]$$

Example: RNA Folding and CKY Parsing

- typical instance of the generalized Viterbi for DAHs
- many variants of CKY ~ various topological ordering
- Nussinov algorithm in RNA is almost identical to CKY but w/o overcounting

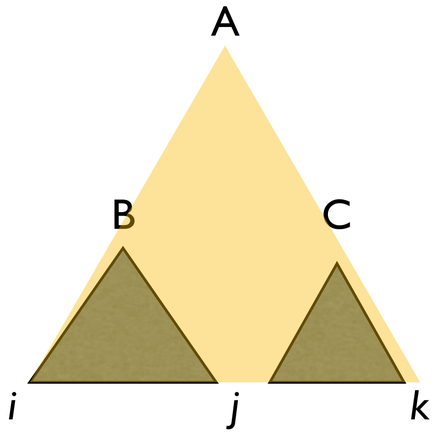


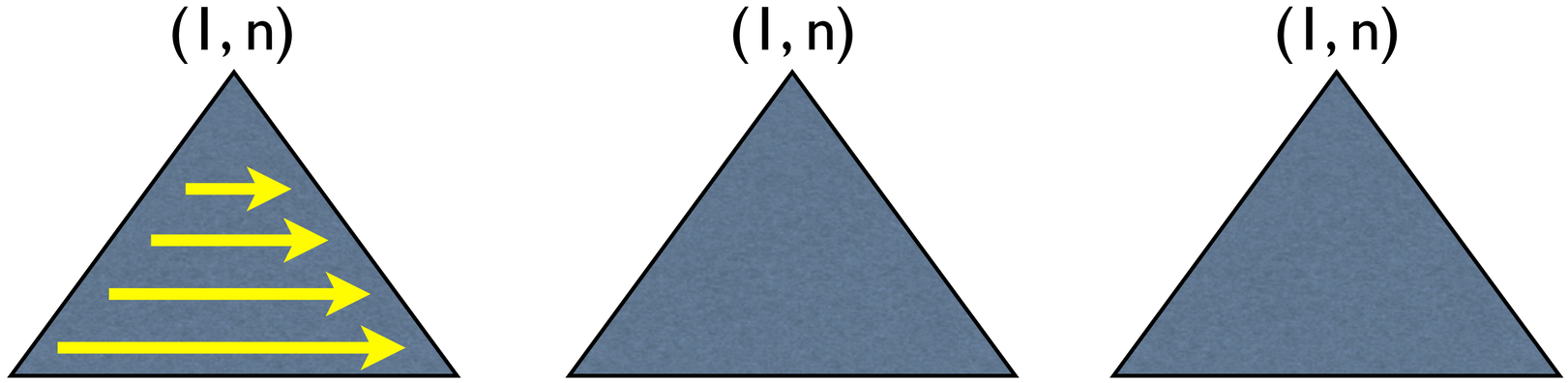


all $O(n^3)$

Example: RNA Folding and CKY Parsing

- typical instance of the generalized Viterbi for DAHs
- many variants of CKY ~ various topological ordering
- Nussinov algorithm in RNA is almost identical to CKY but w/o overcounting



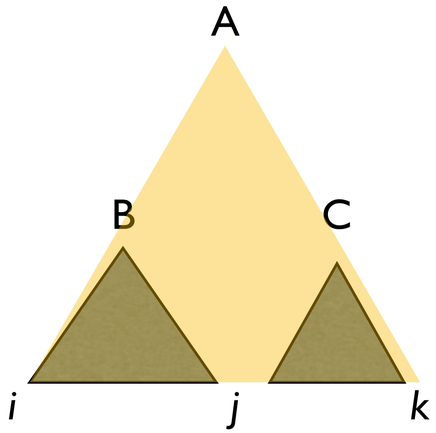


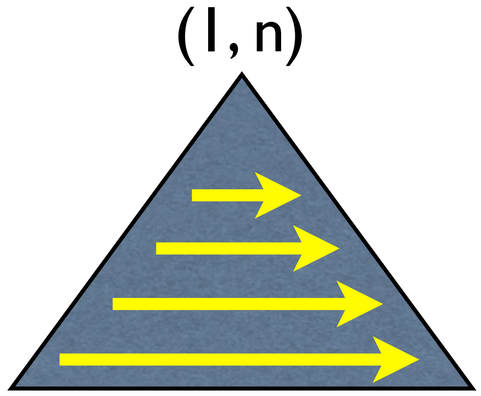
bottom-up

all $O(n^3)$

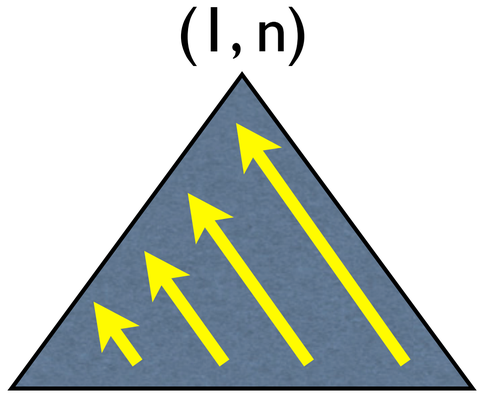
Example: RNA Folding and CKY Parsing

- typical instance of the generalized Viterbi for DAHs
- many variants of CKY ~ various topological ordering
- Nussinov algorithm in RNA is almost identical to CKY but w/o overcounting

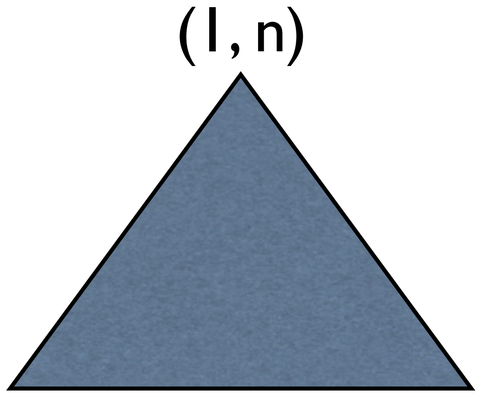




bottom-up



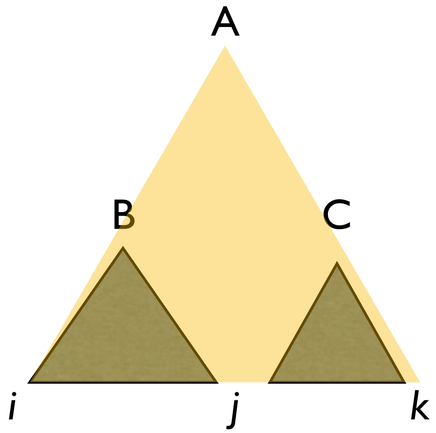
left-to-right

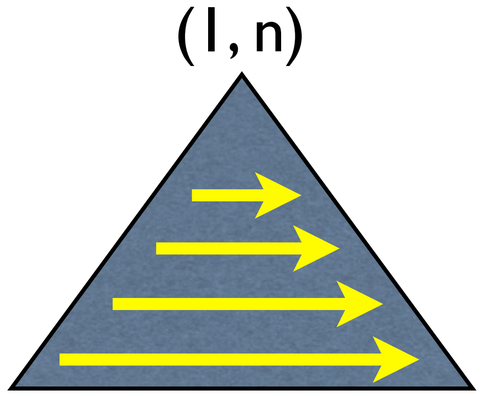


all $O(n^3)$

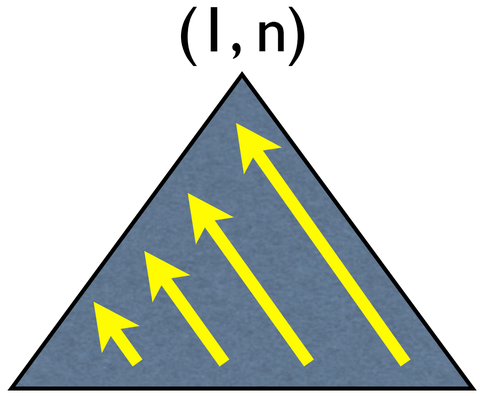
Example: RNA Folding and CKY Parsing

- typical instance of the generalized Viterbi for DAHs
- many variants of CKY ~ various topological ordering
- Nussinov algorithm in RNA is almost identical to CKY but w/o overcounting

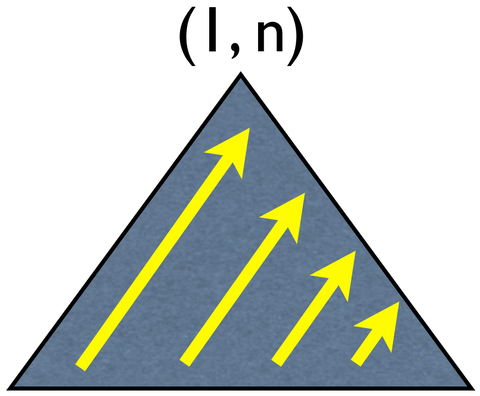




bottom-up



left-to-right

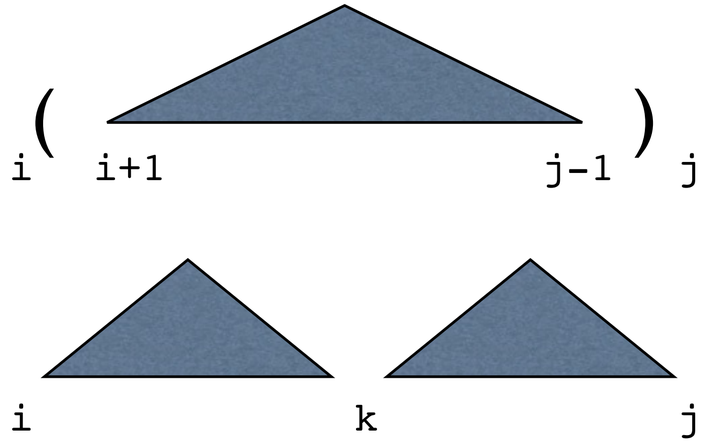


right-to-left

all $O(n^3)$

Example: RNA Folding as CKY Parsing

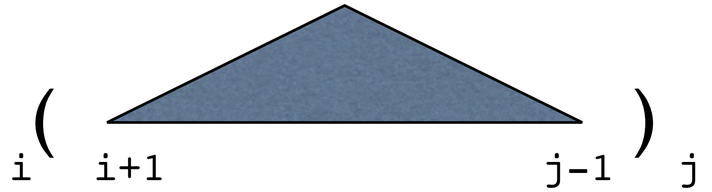
- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)

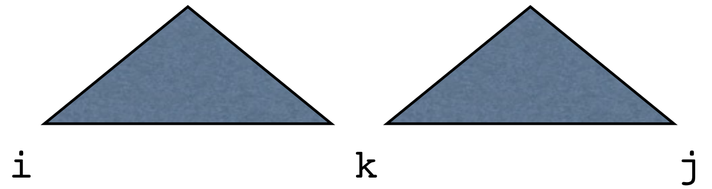


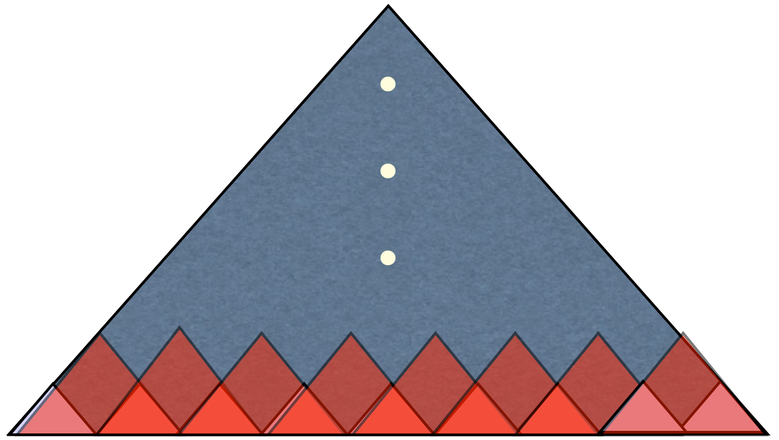
A C A G U

Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



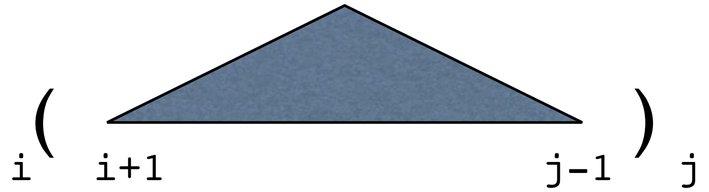


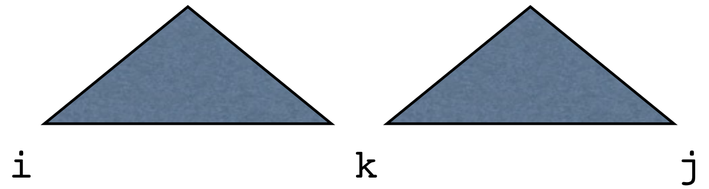


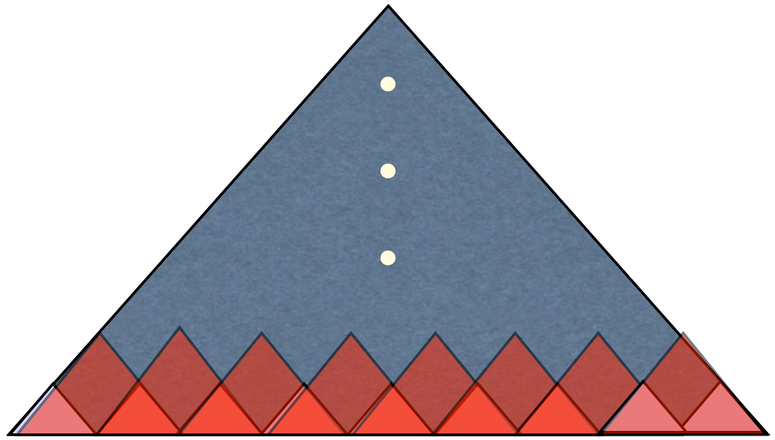
A C A G U

Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



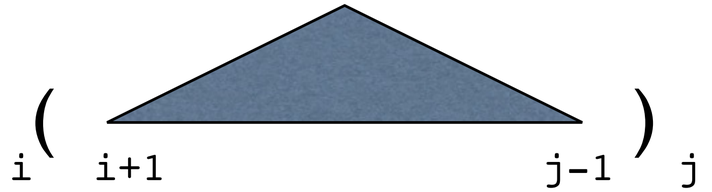


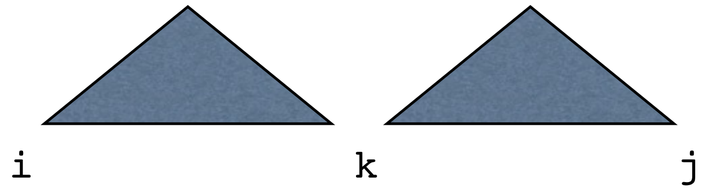


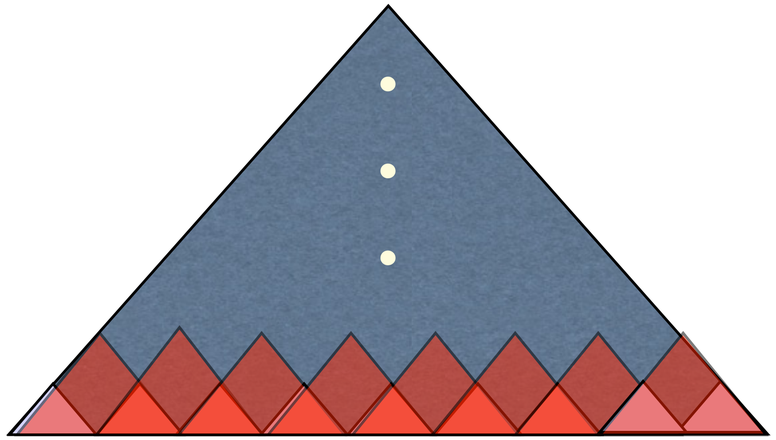
A C A G U

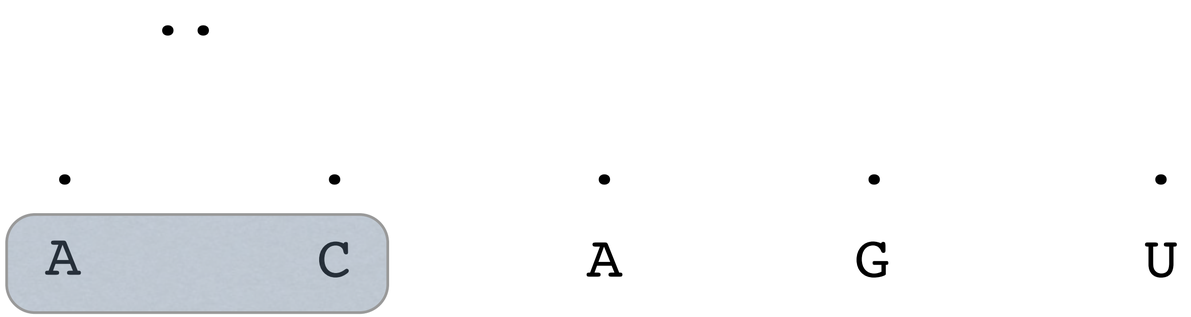
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



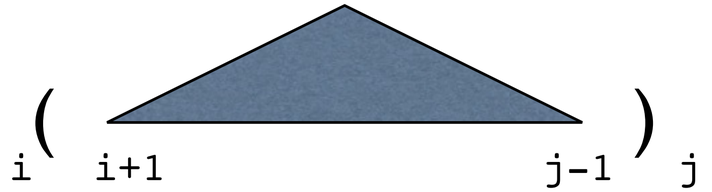


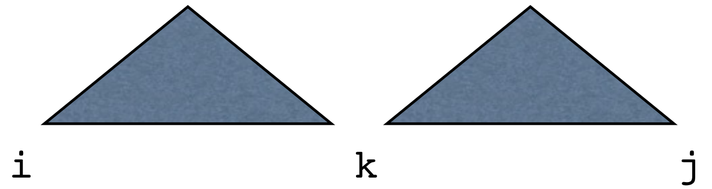


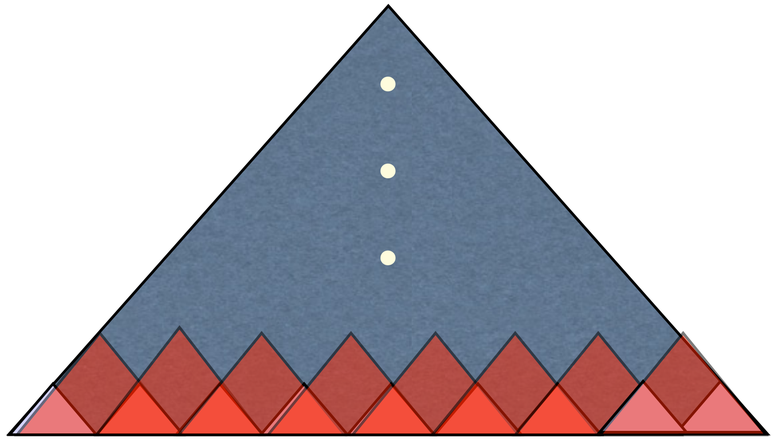


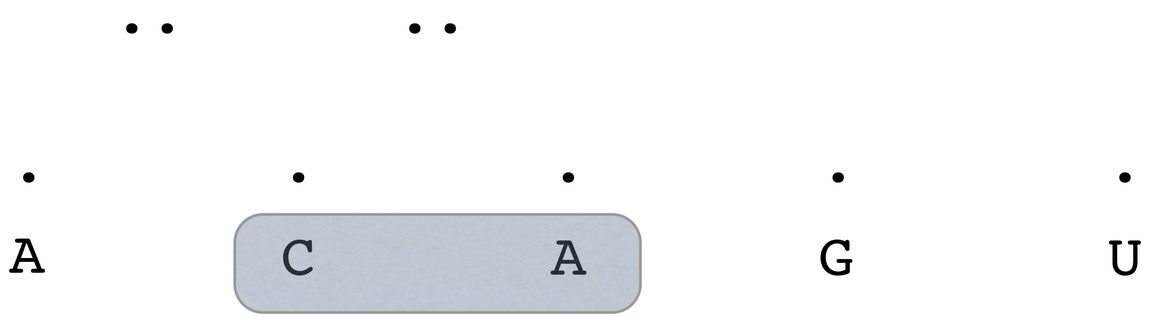
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



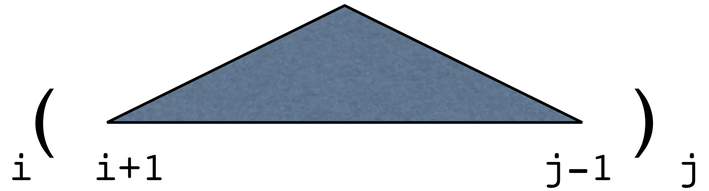


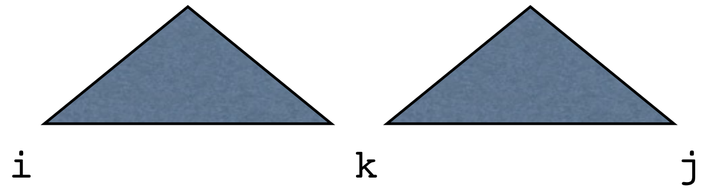


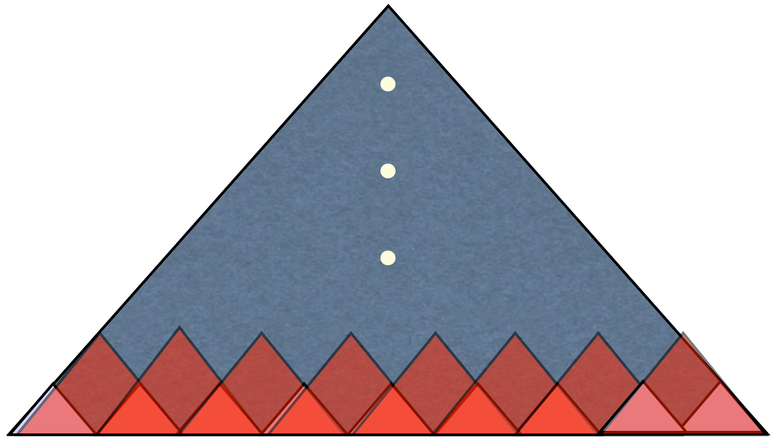


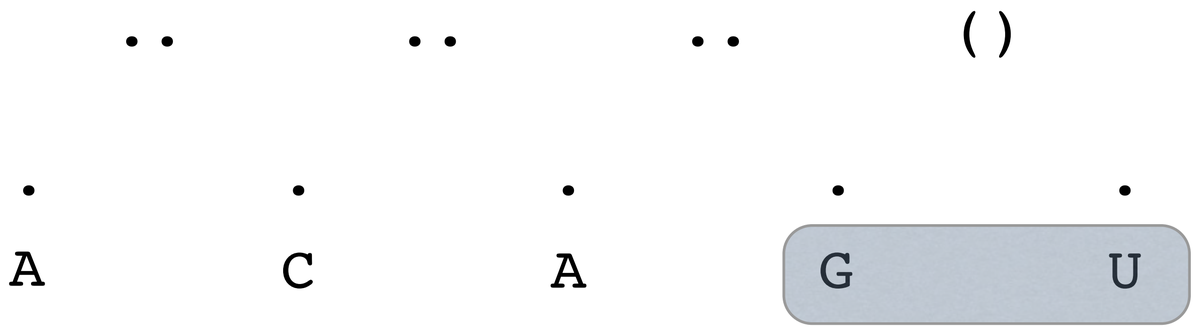
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



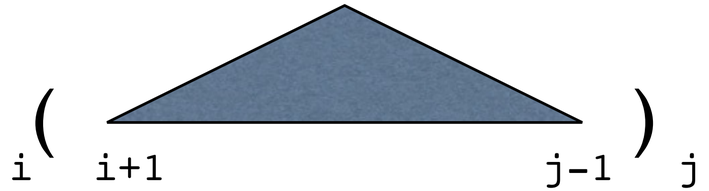


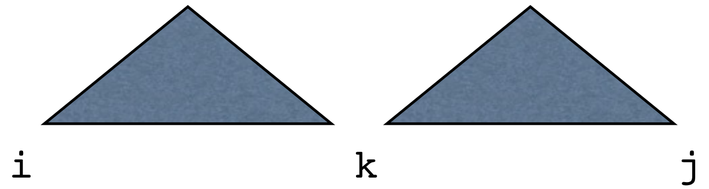


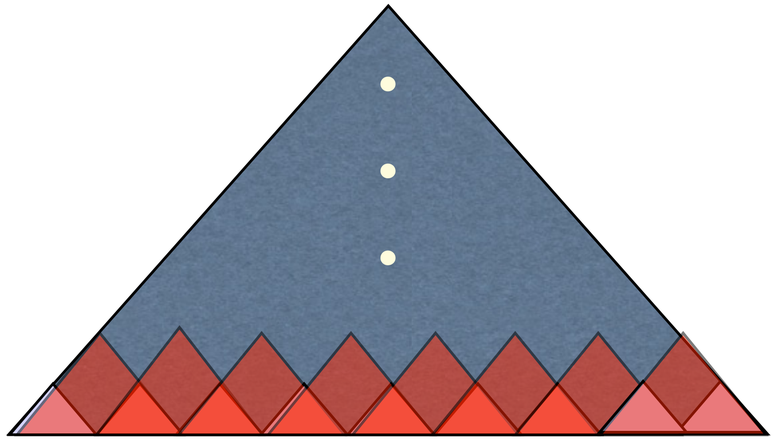


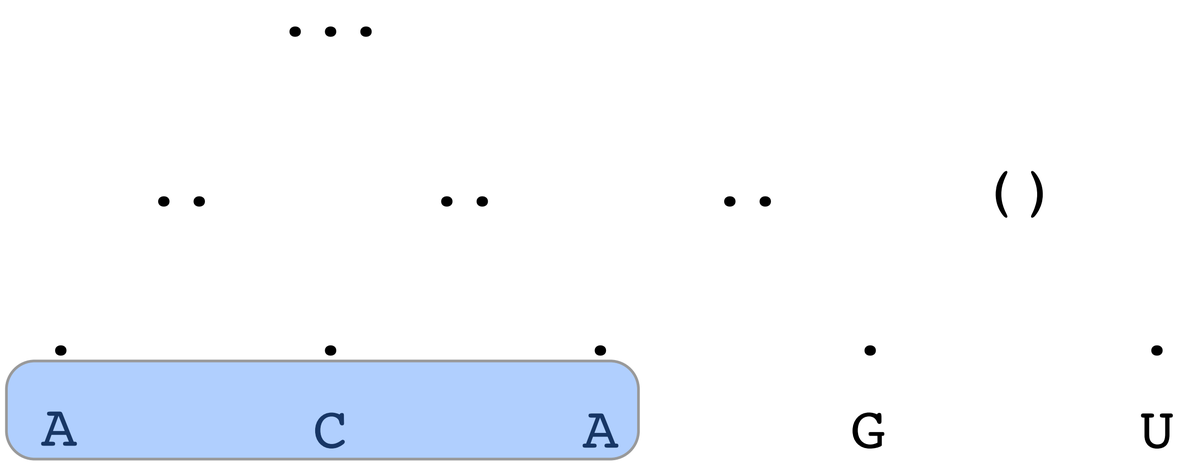
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



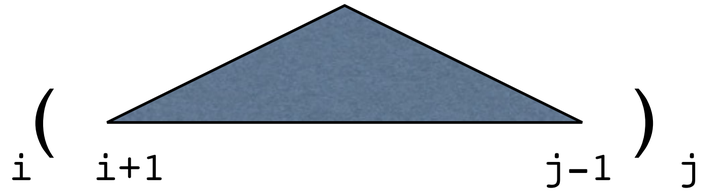


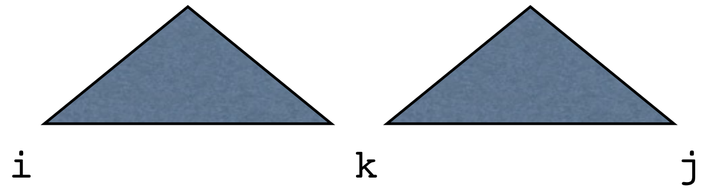


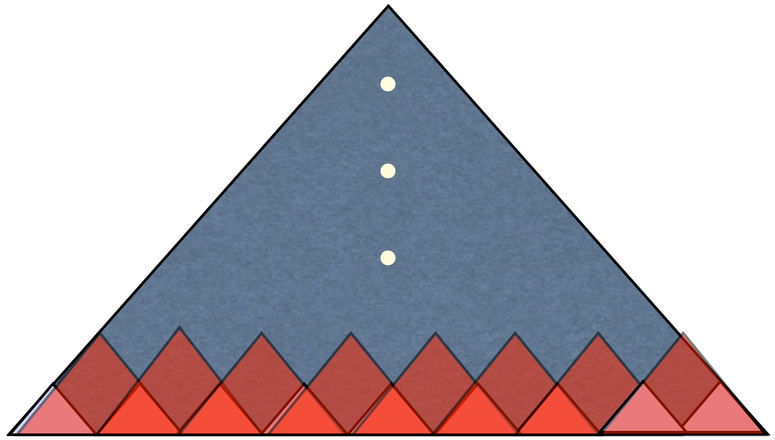


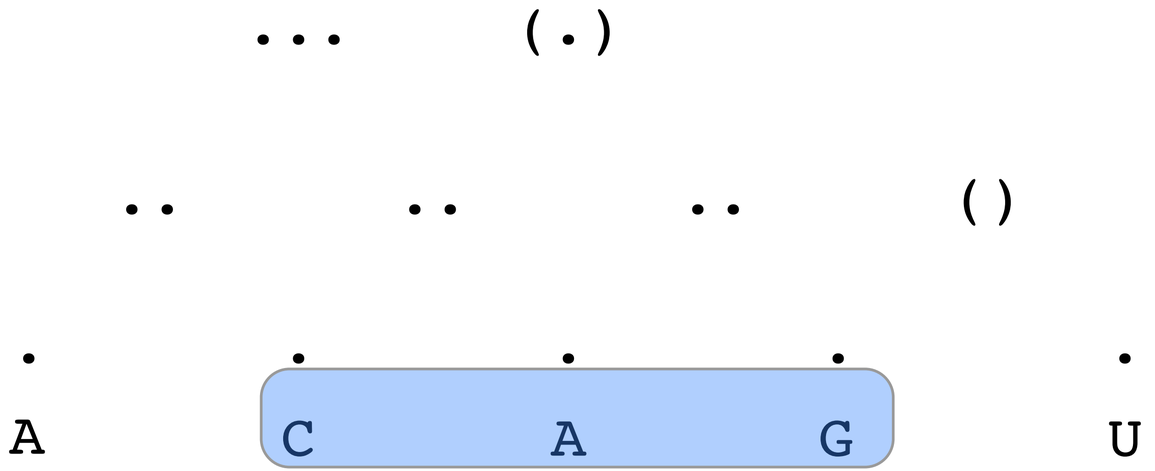
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



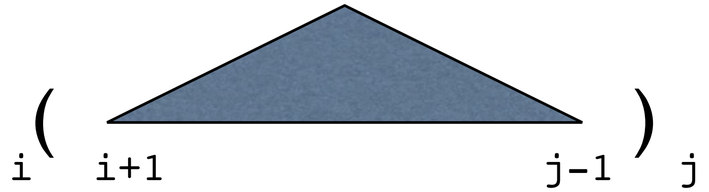


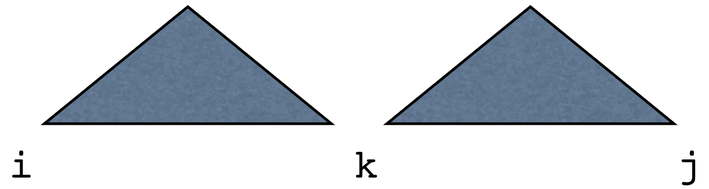


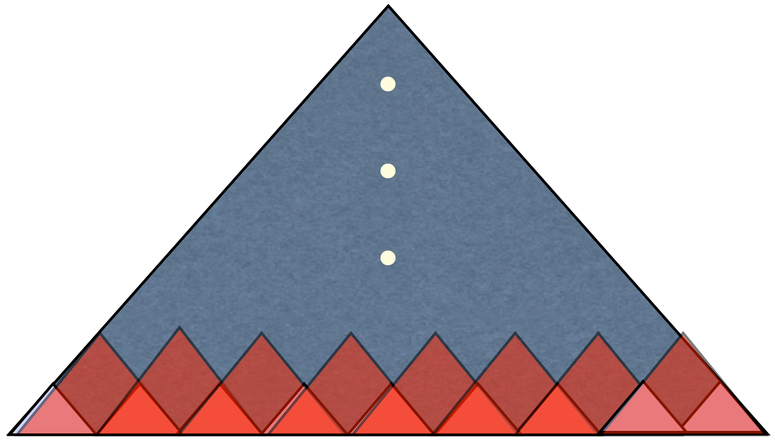


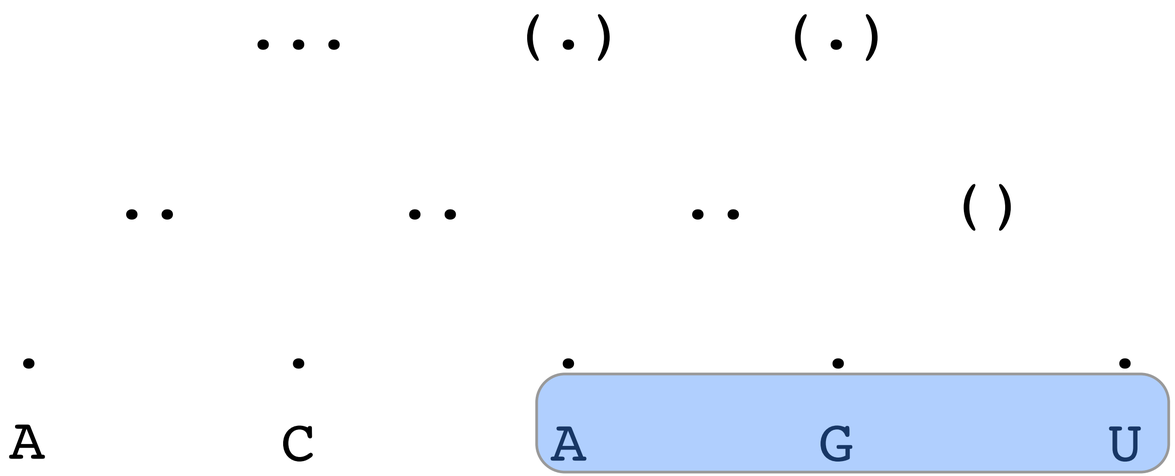
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



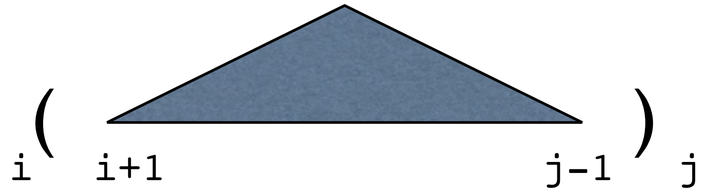


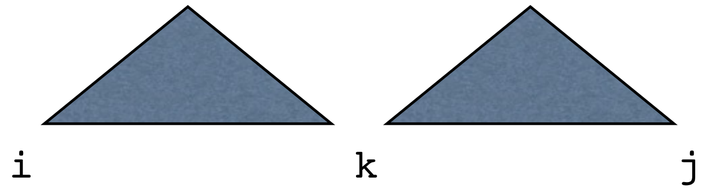


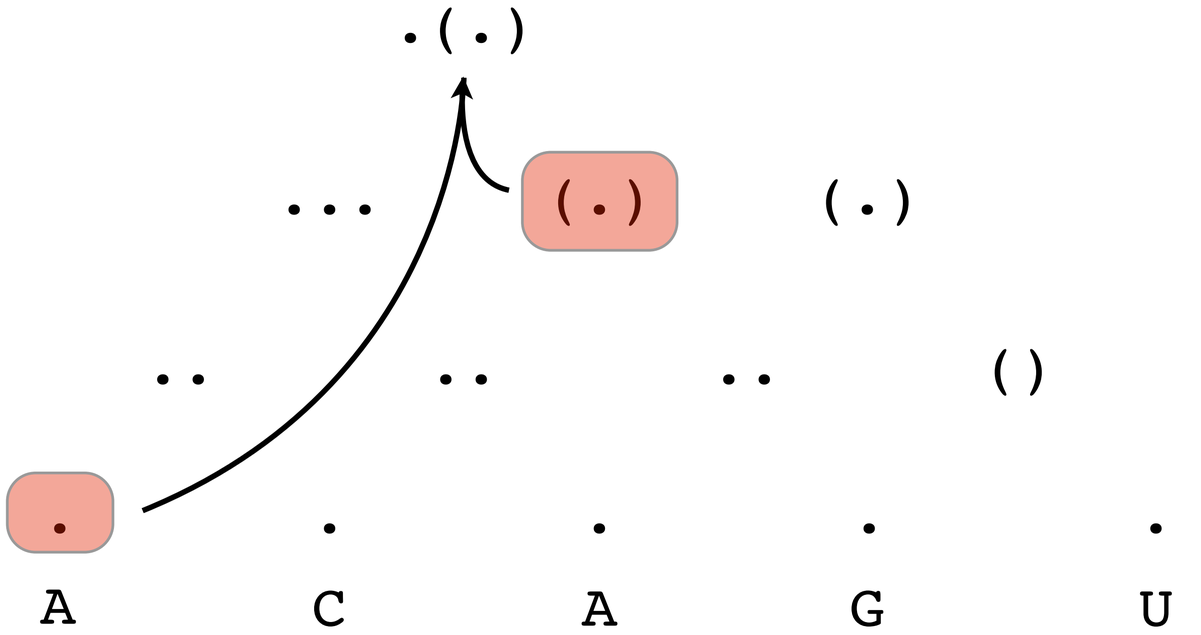


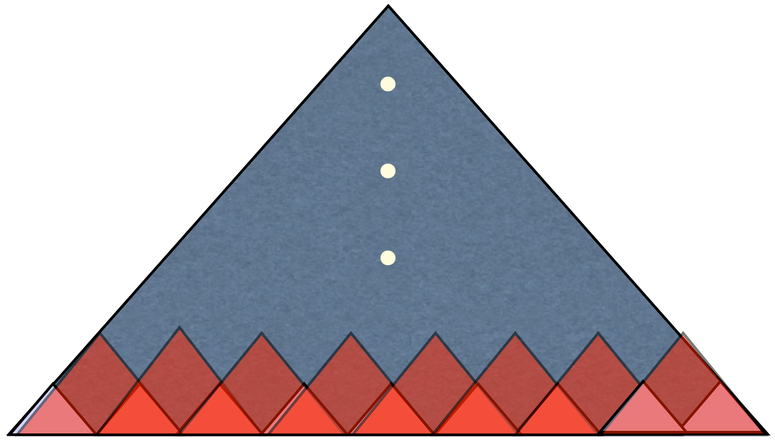
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



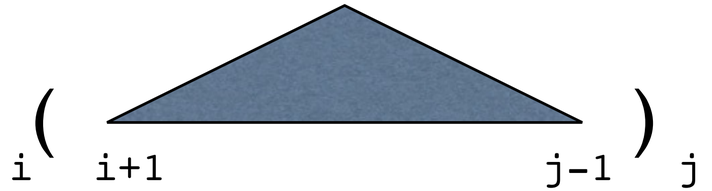


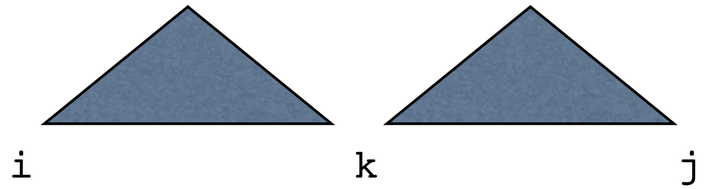


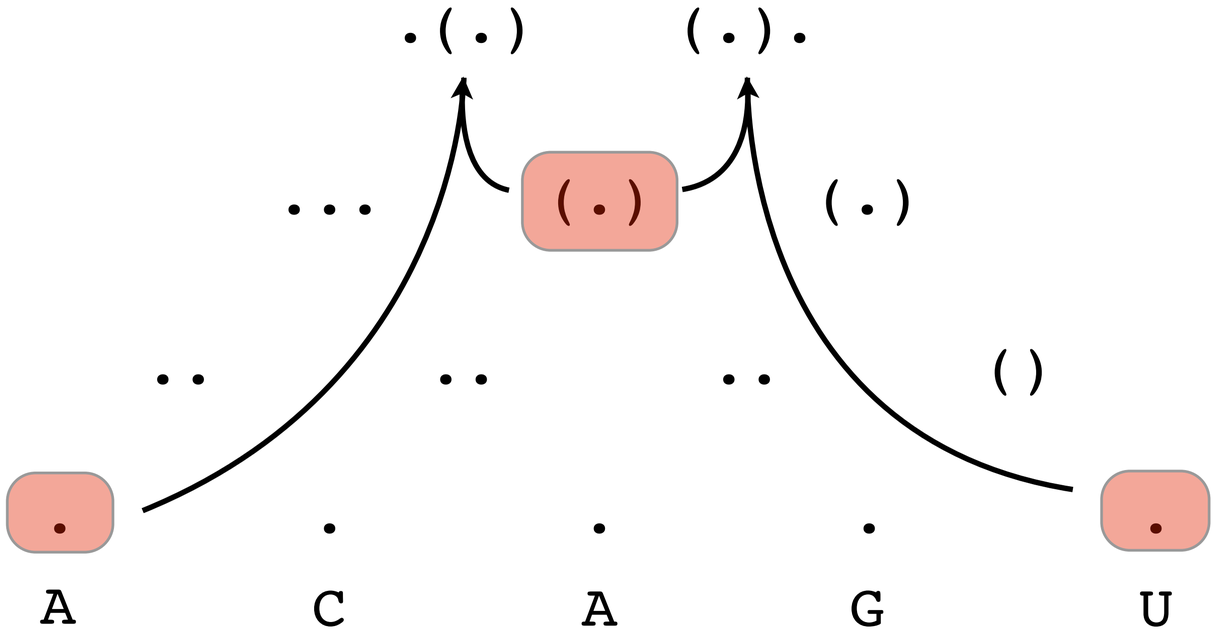


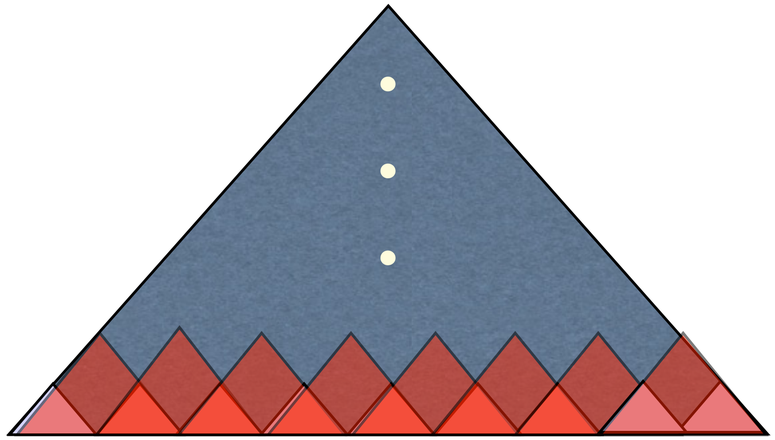
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



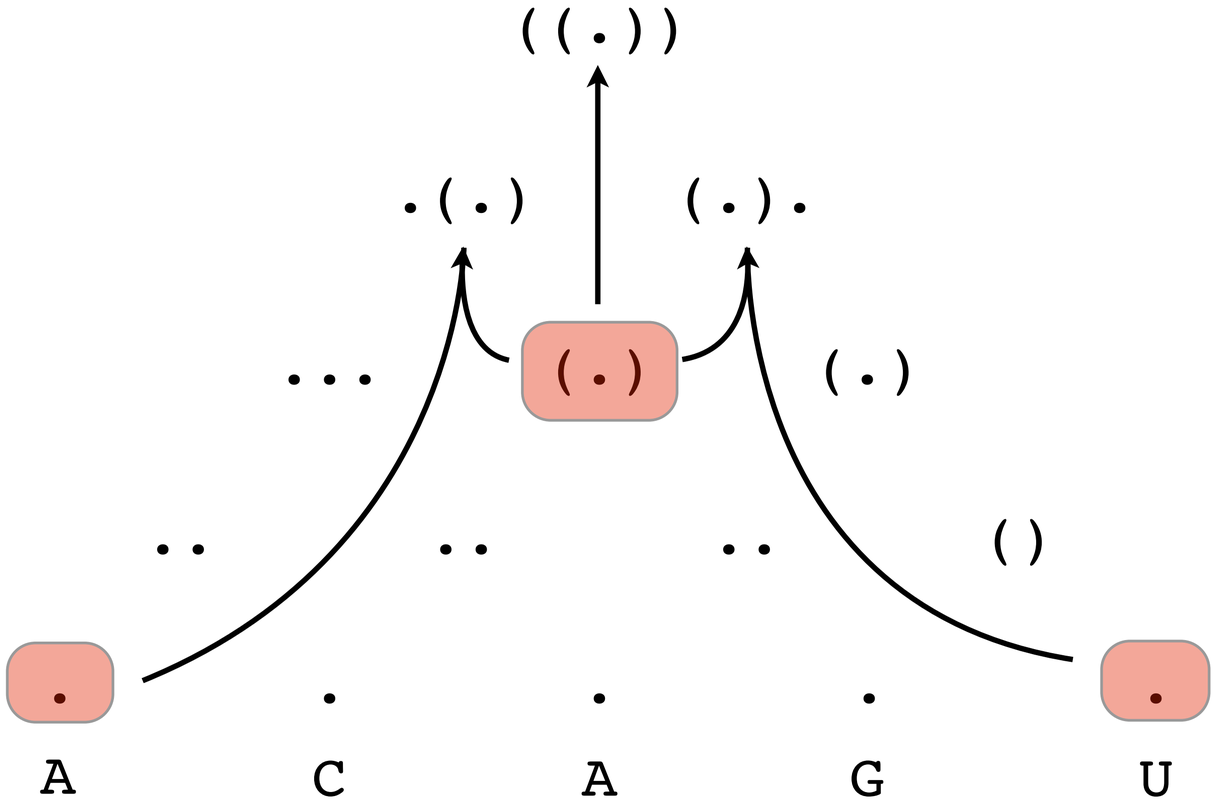


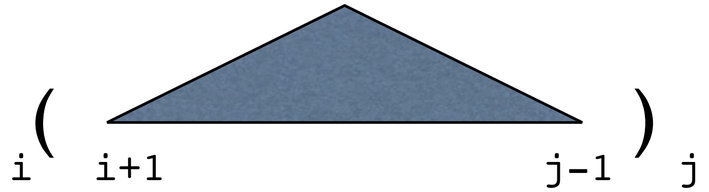


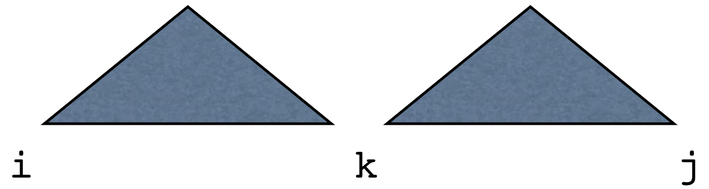


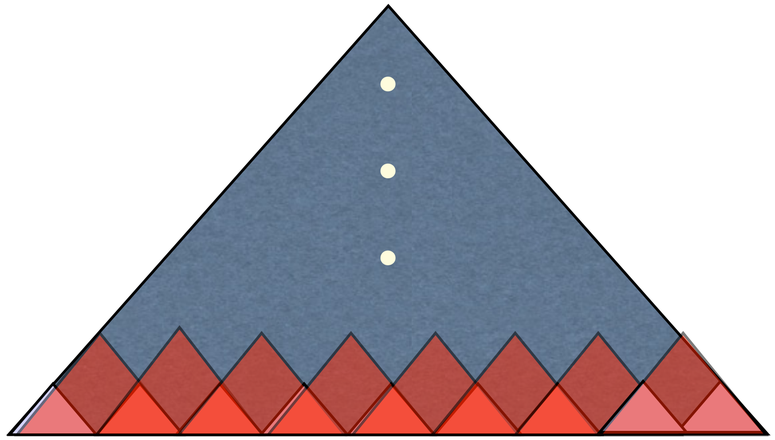
Example: RNA Folding as CKY Parsing

- Dynamic Programming — $O(n^3)$
  - bottom-up CKY parsing
  - example: maximize # of pairs (A-U, G-C, or G-U)



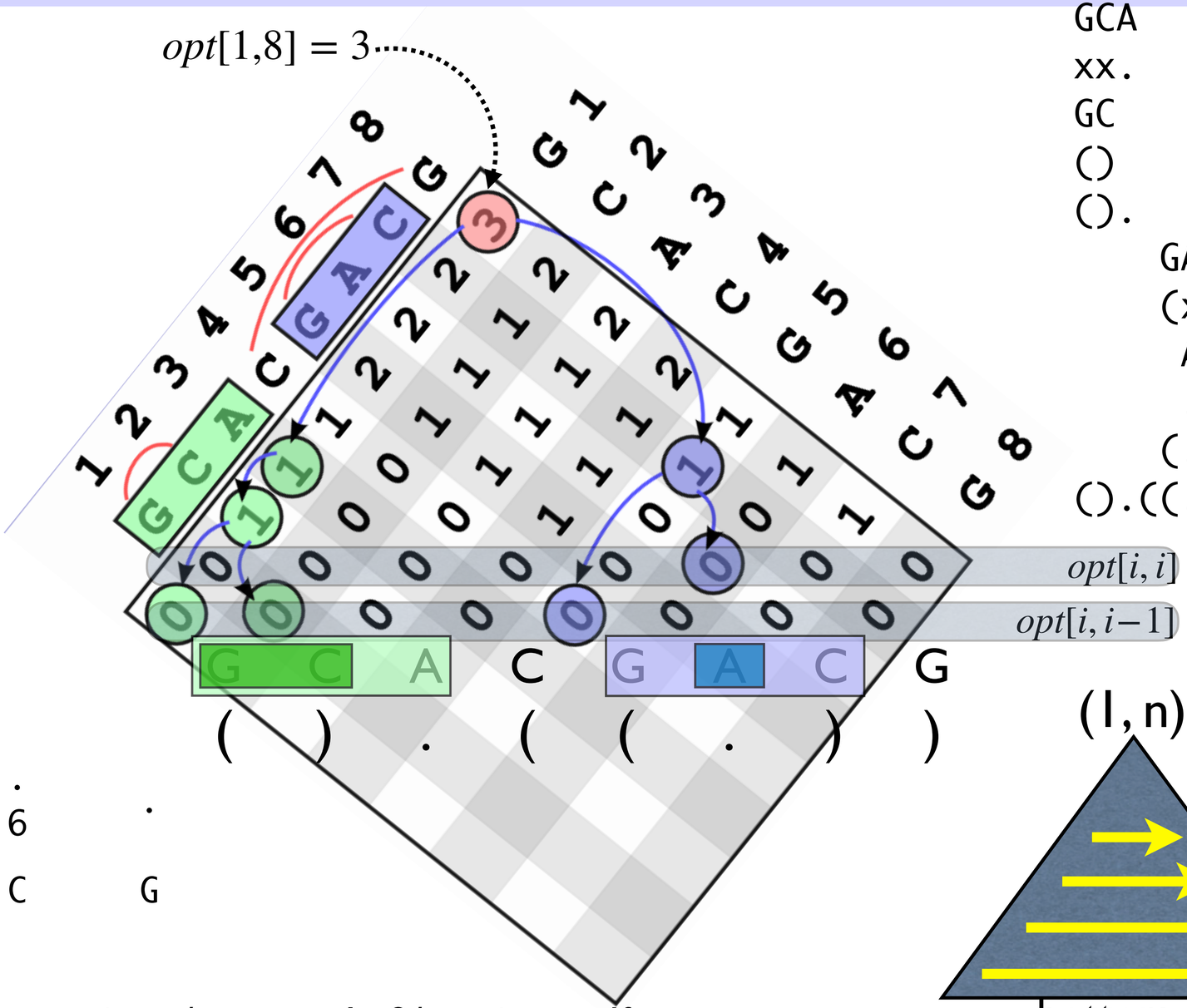






RNA Folding Example: Nussinov

```
12345678
GCACGACG
xxx(xxx)
GCA
xx.
GC
()
().
    GAC
    (x)
     A
     .
    (.)
().((.))
```

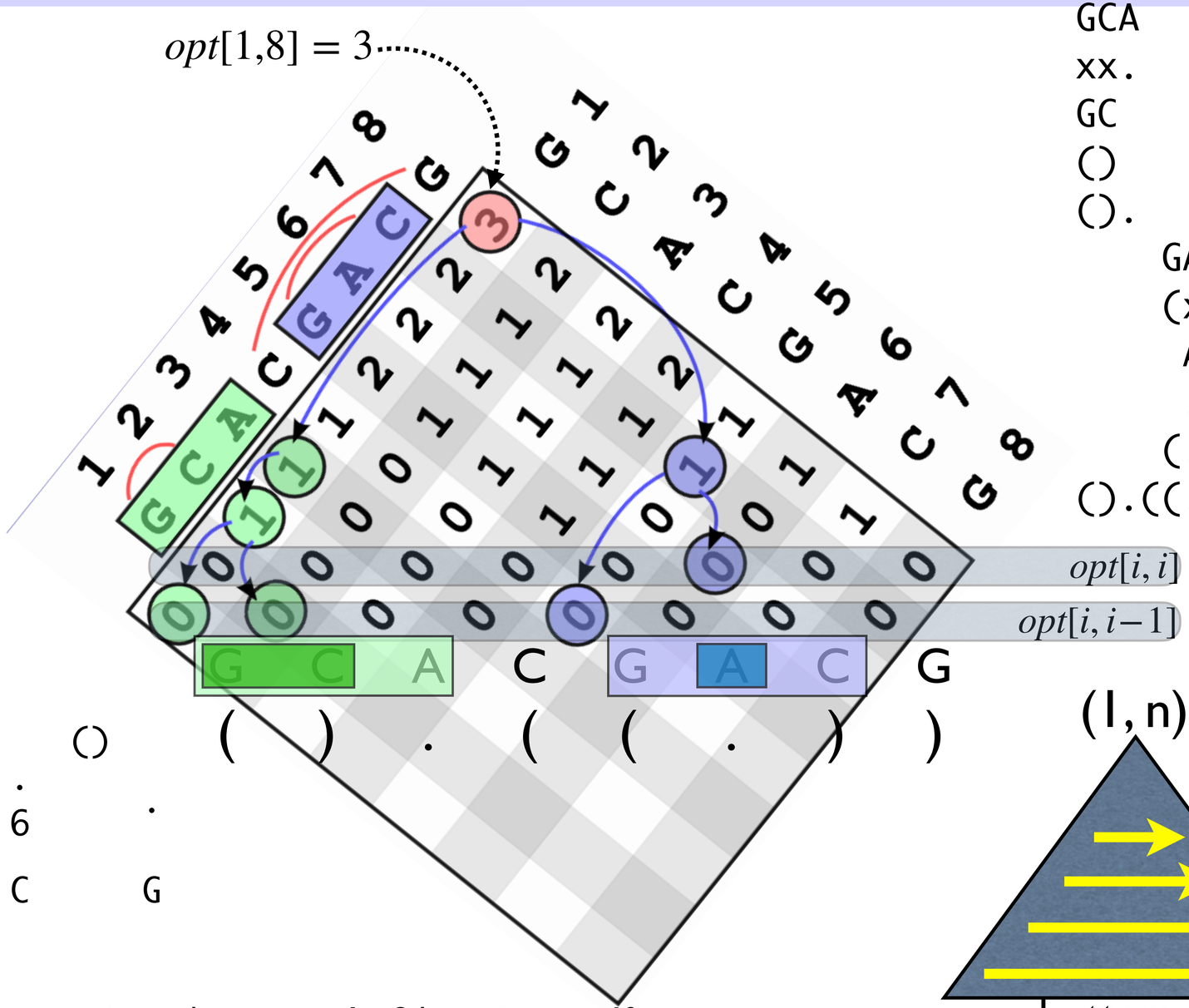



```
.  .  .  .  .  .  .  .
      6
G  C  A  C  G  A  C  G
```

bottom-up

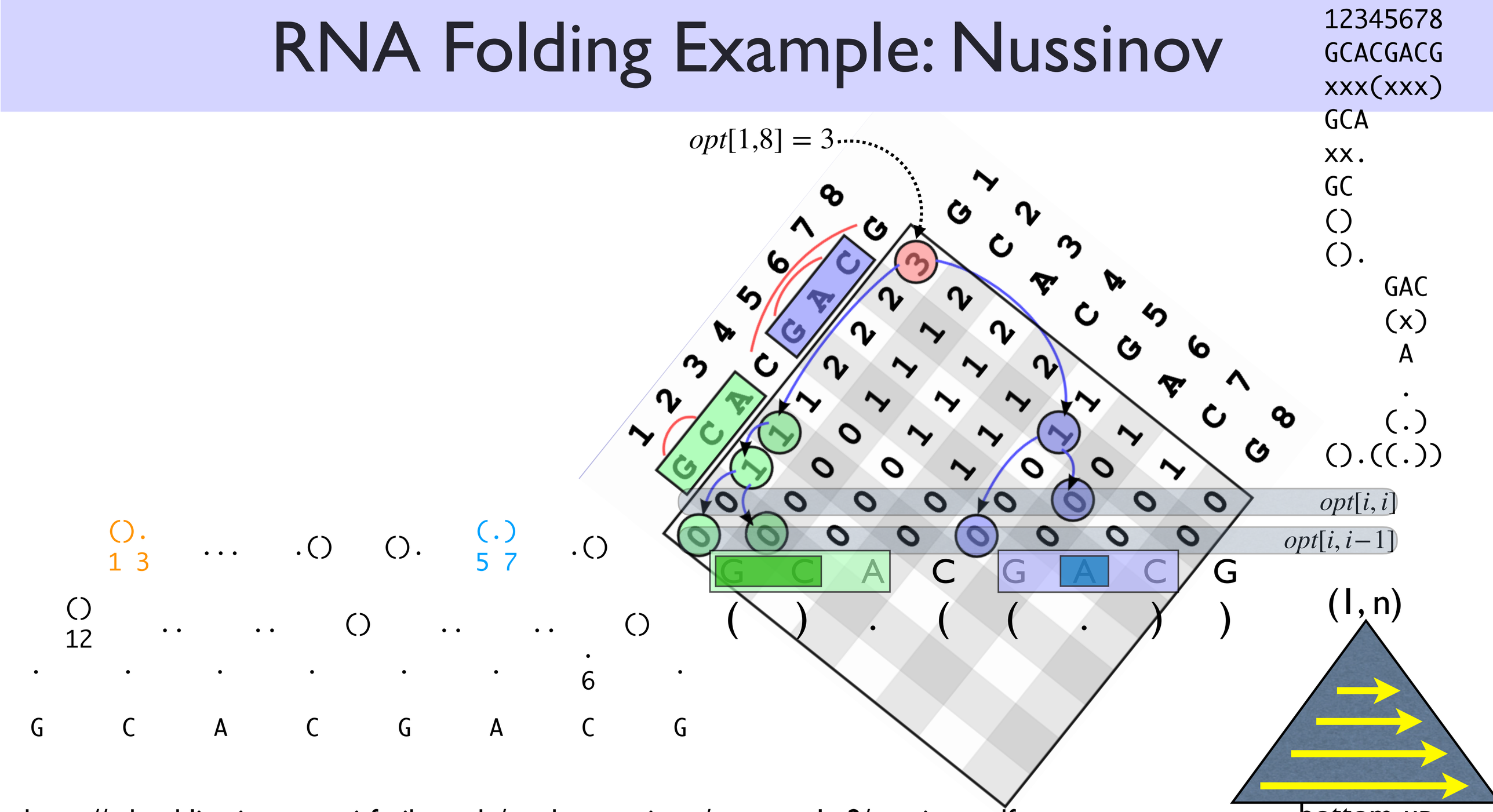
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



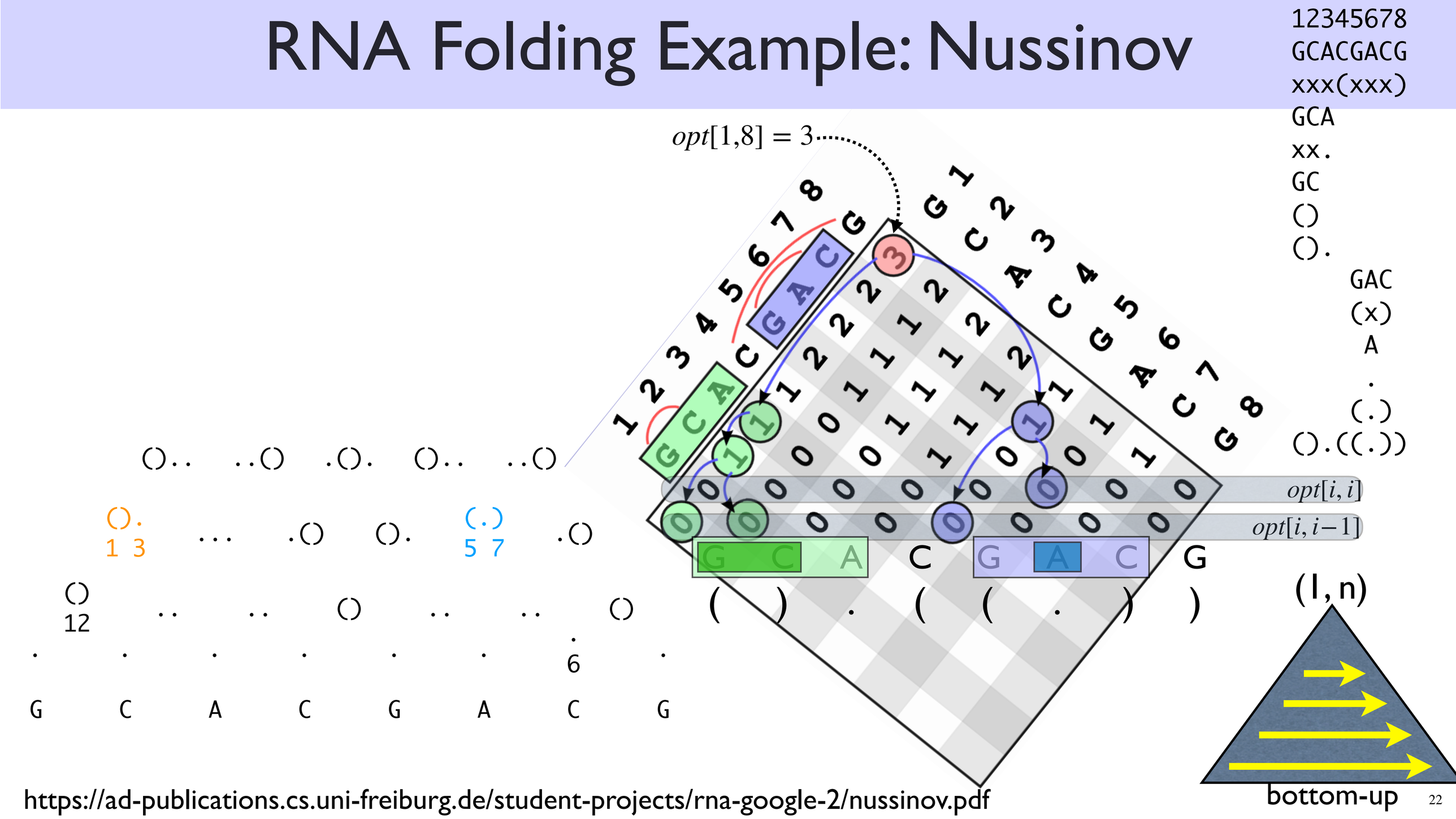
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



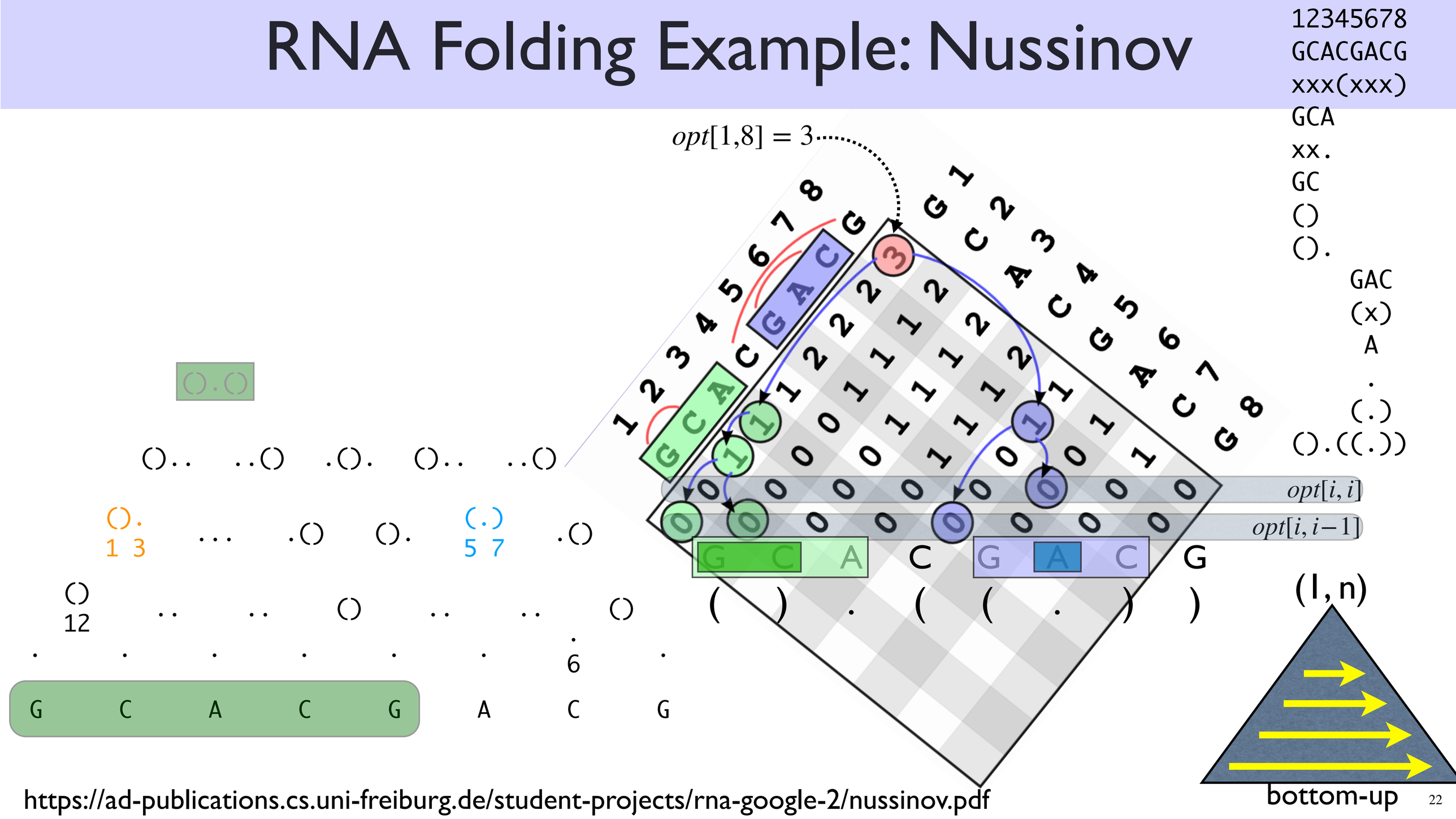
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



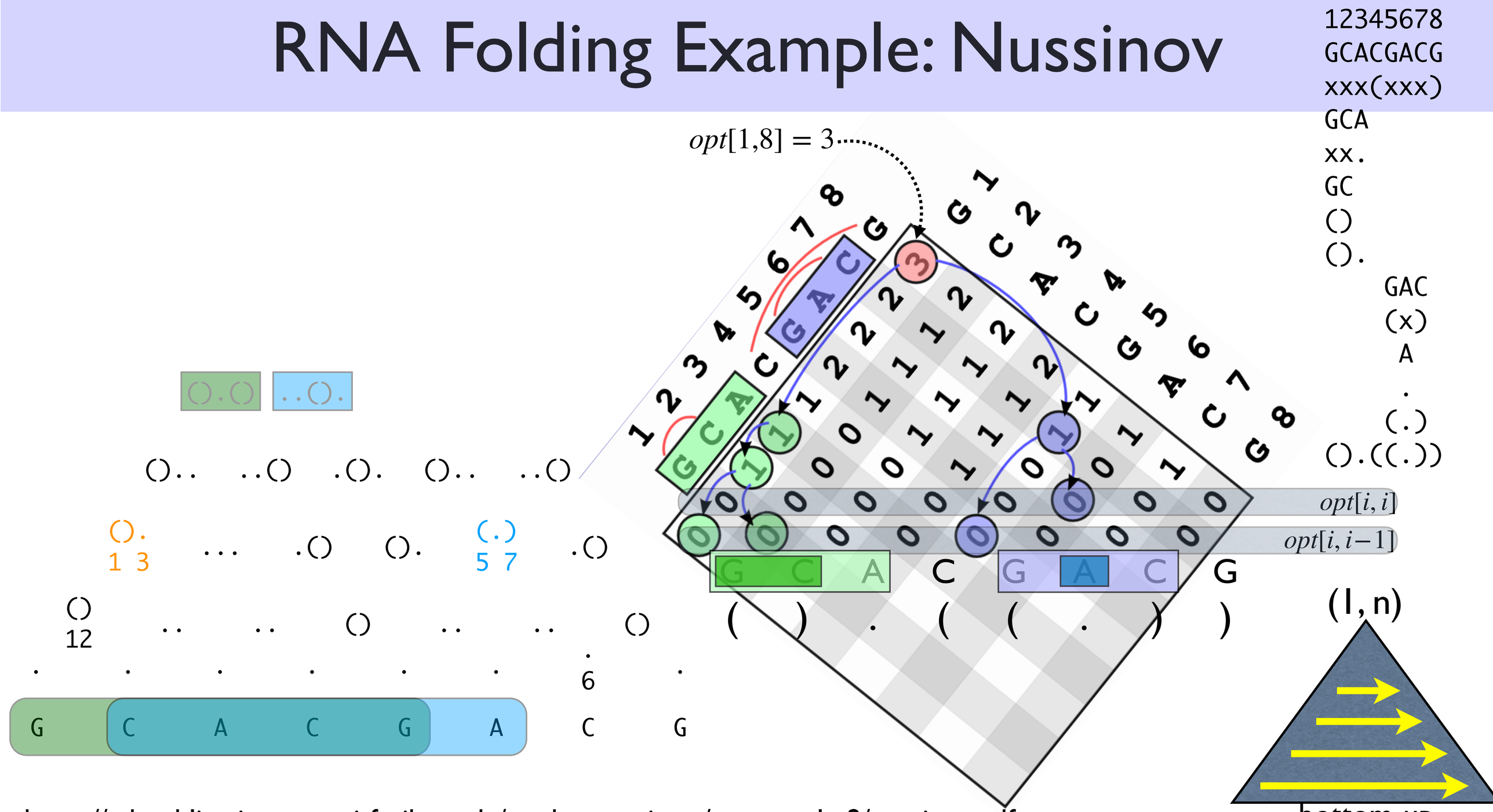
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



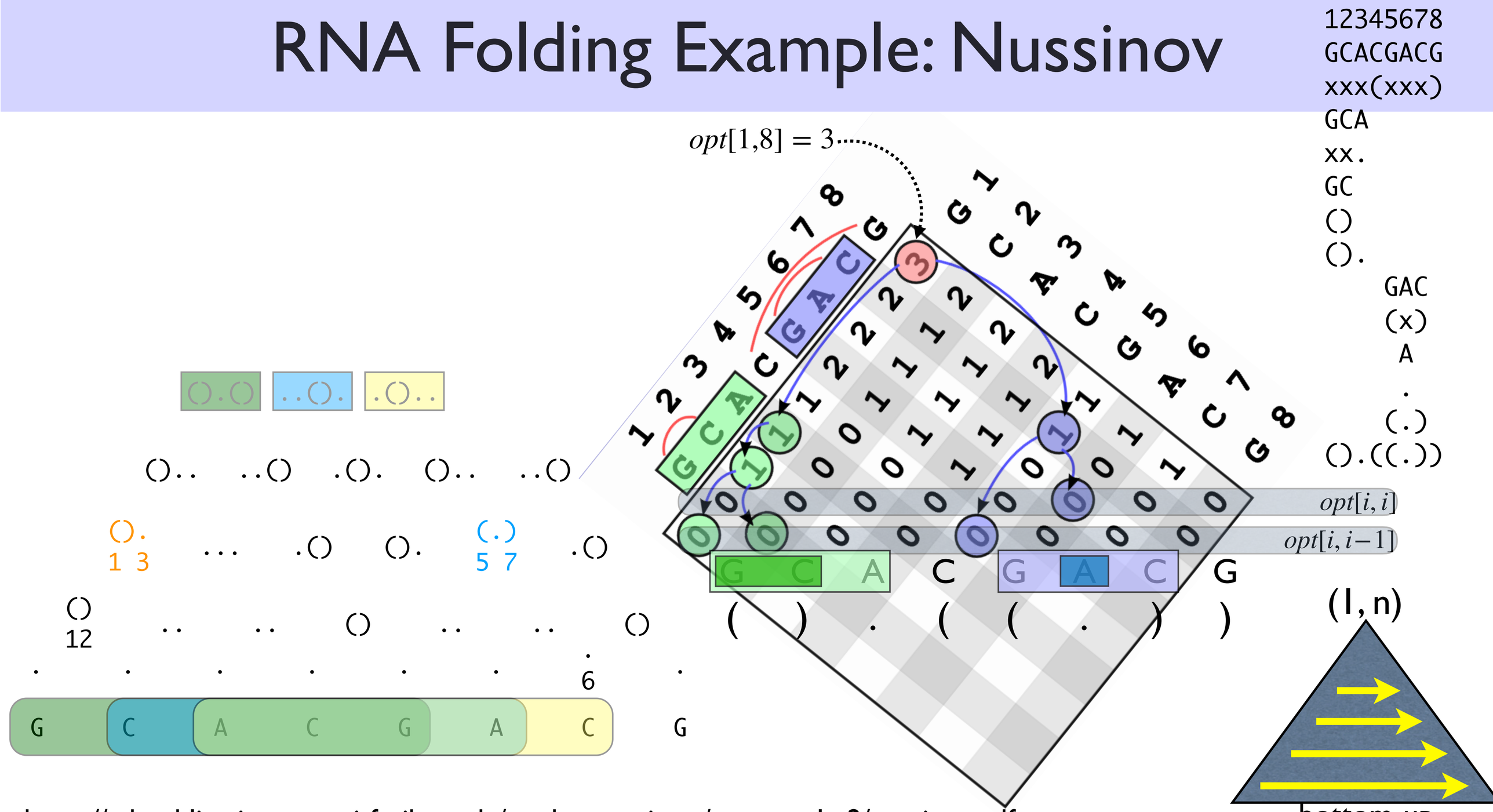
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



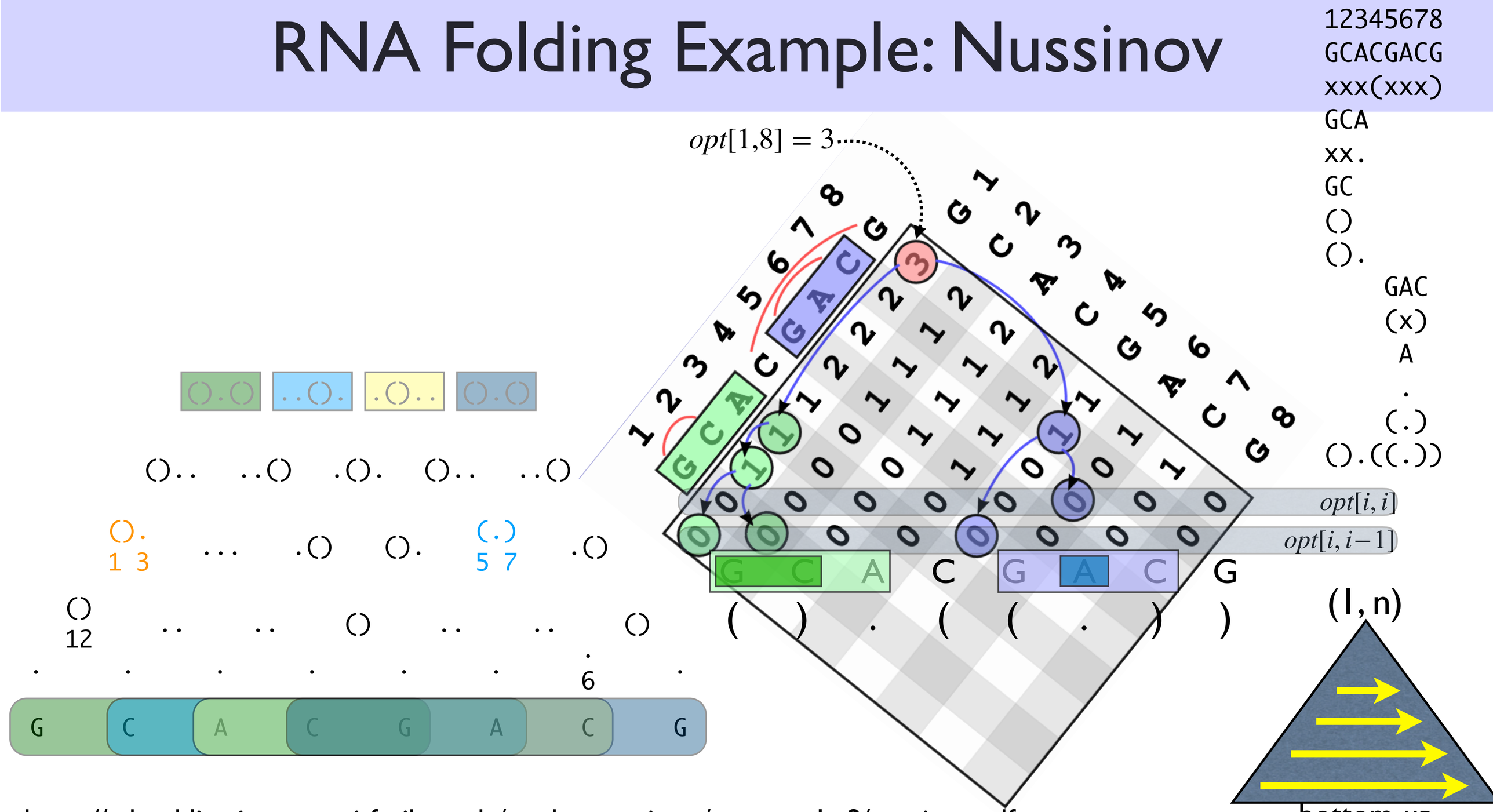
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



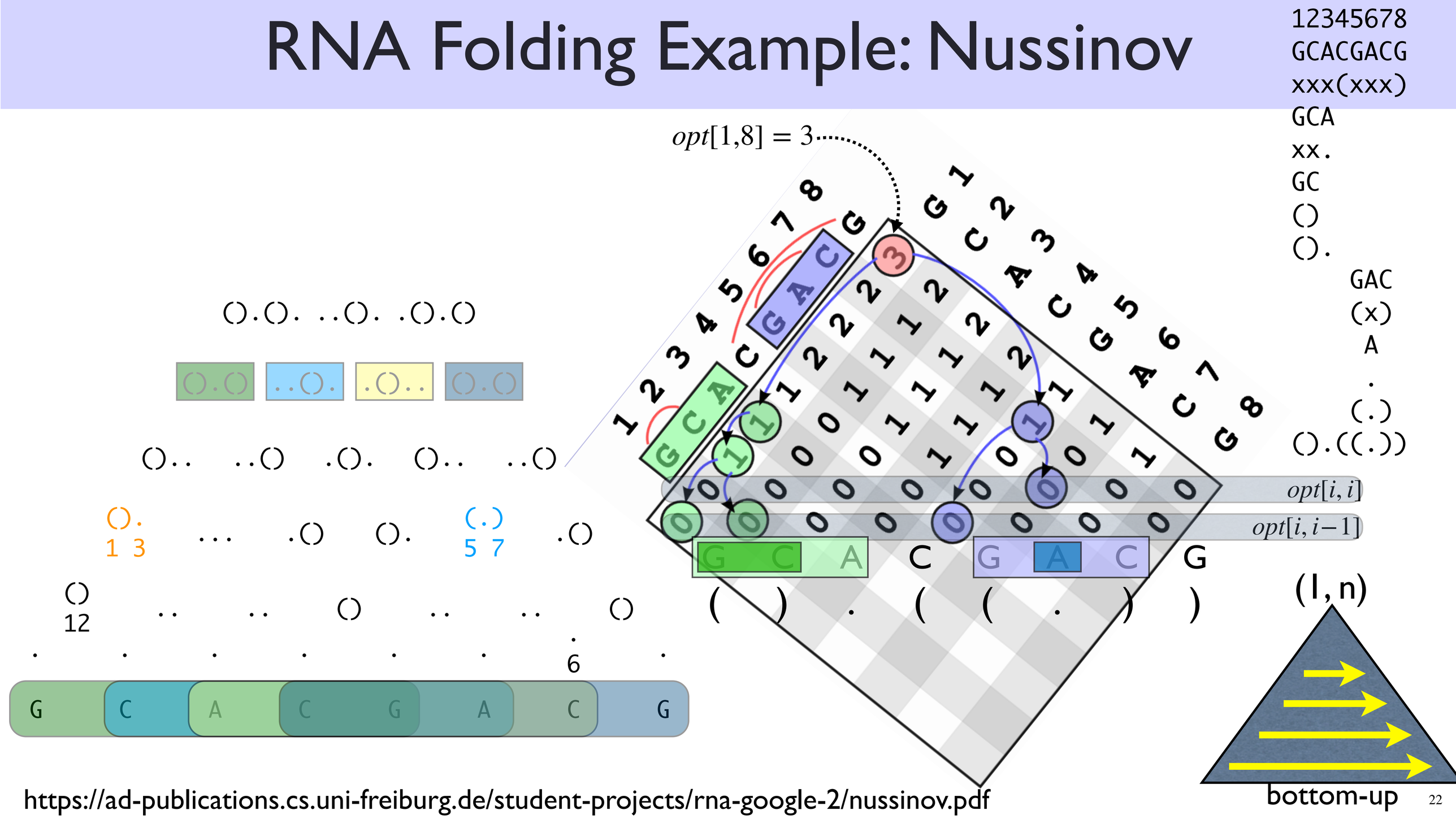
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



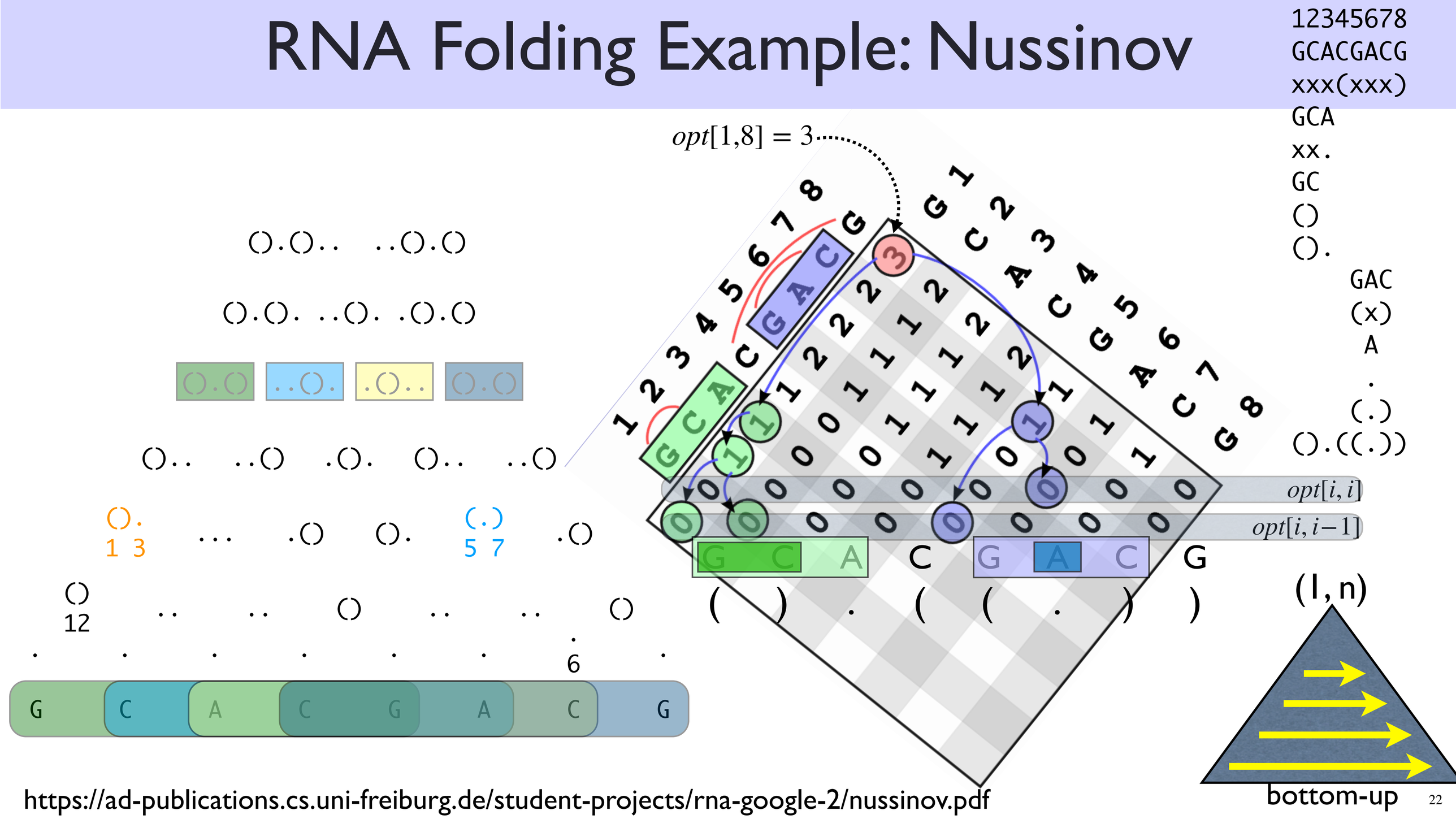
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov



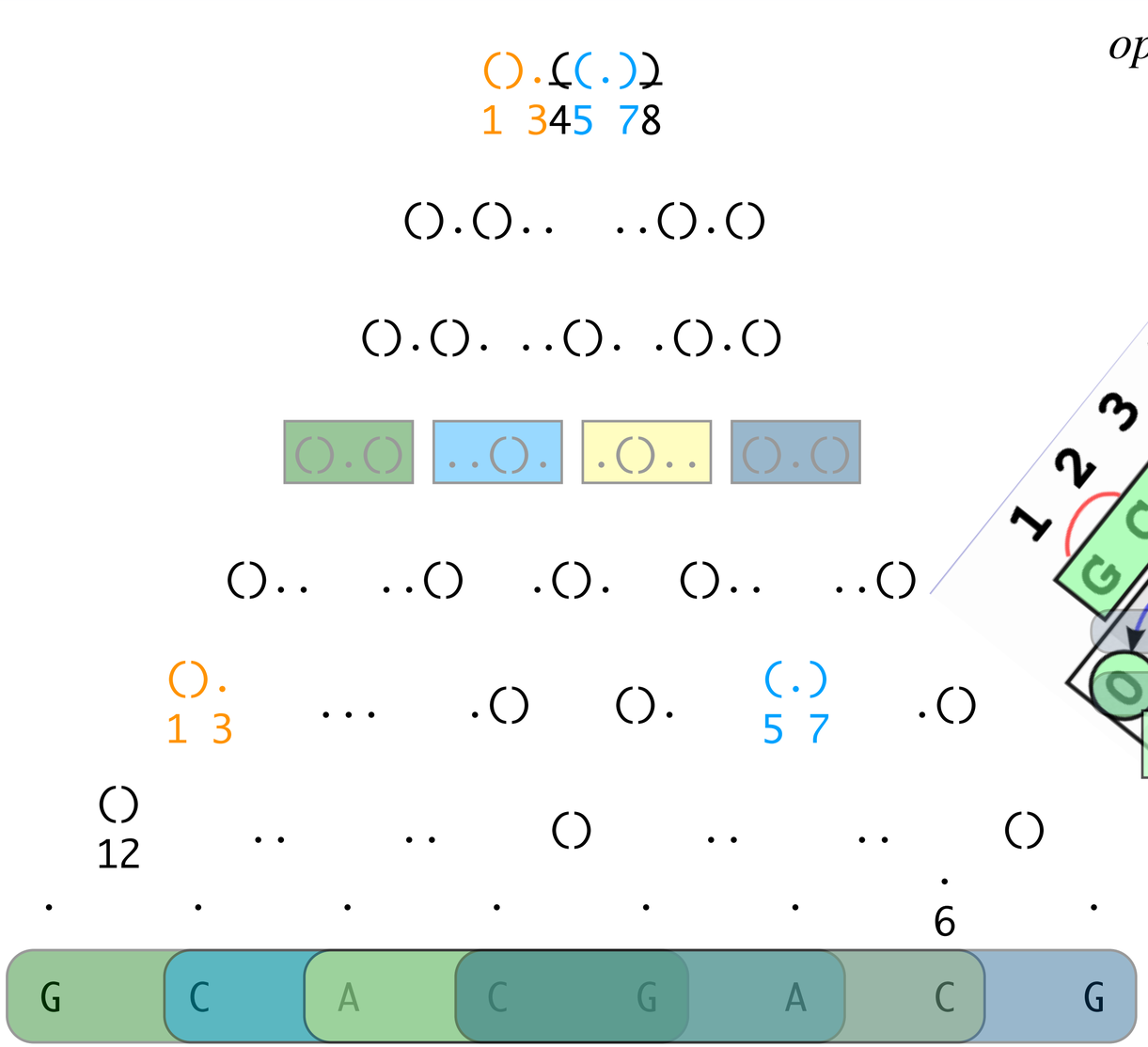
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

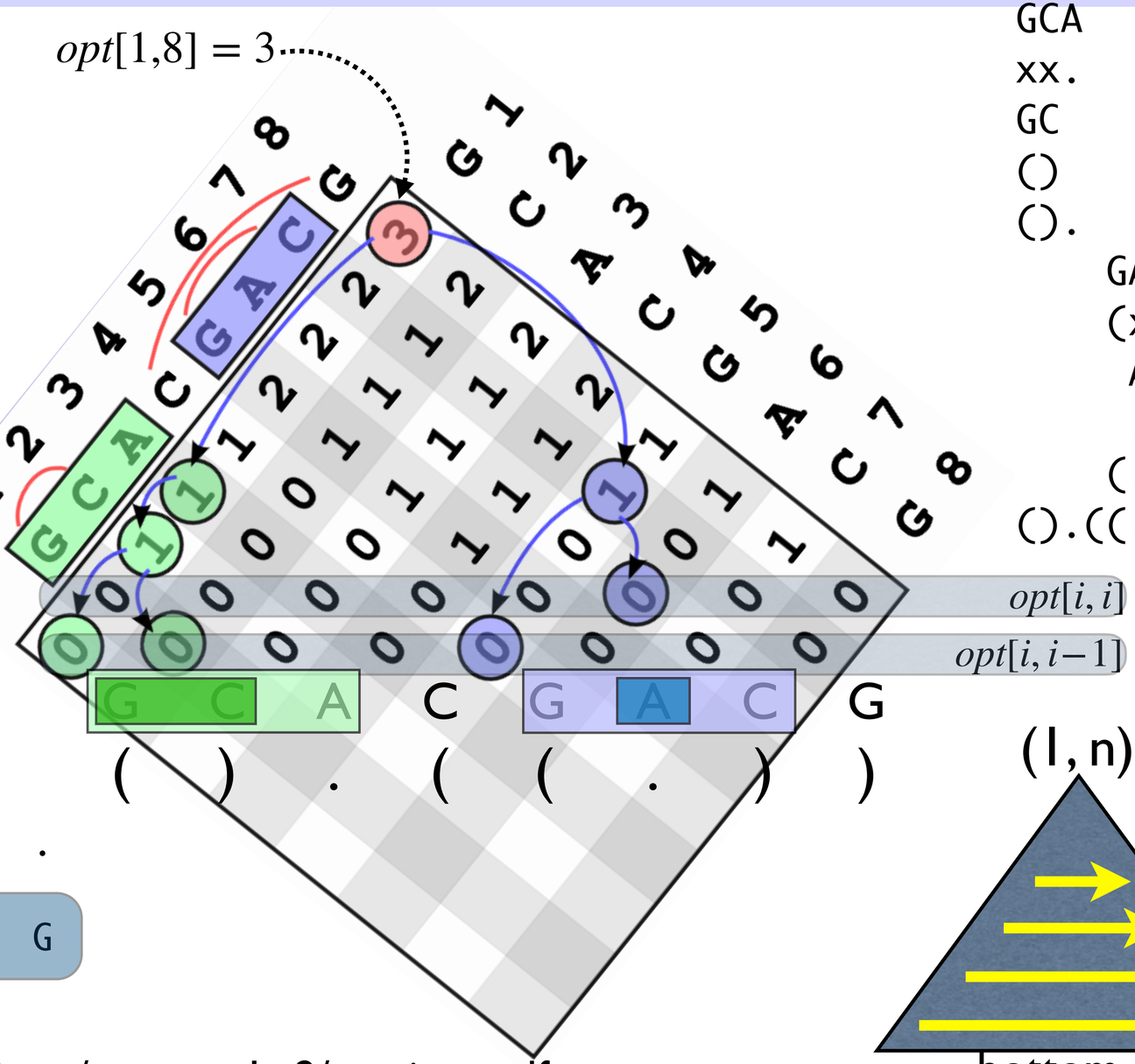
RNA Folding Example: Nussinov



https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov





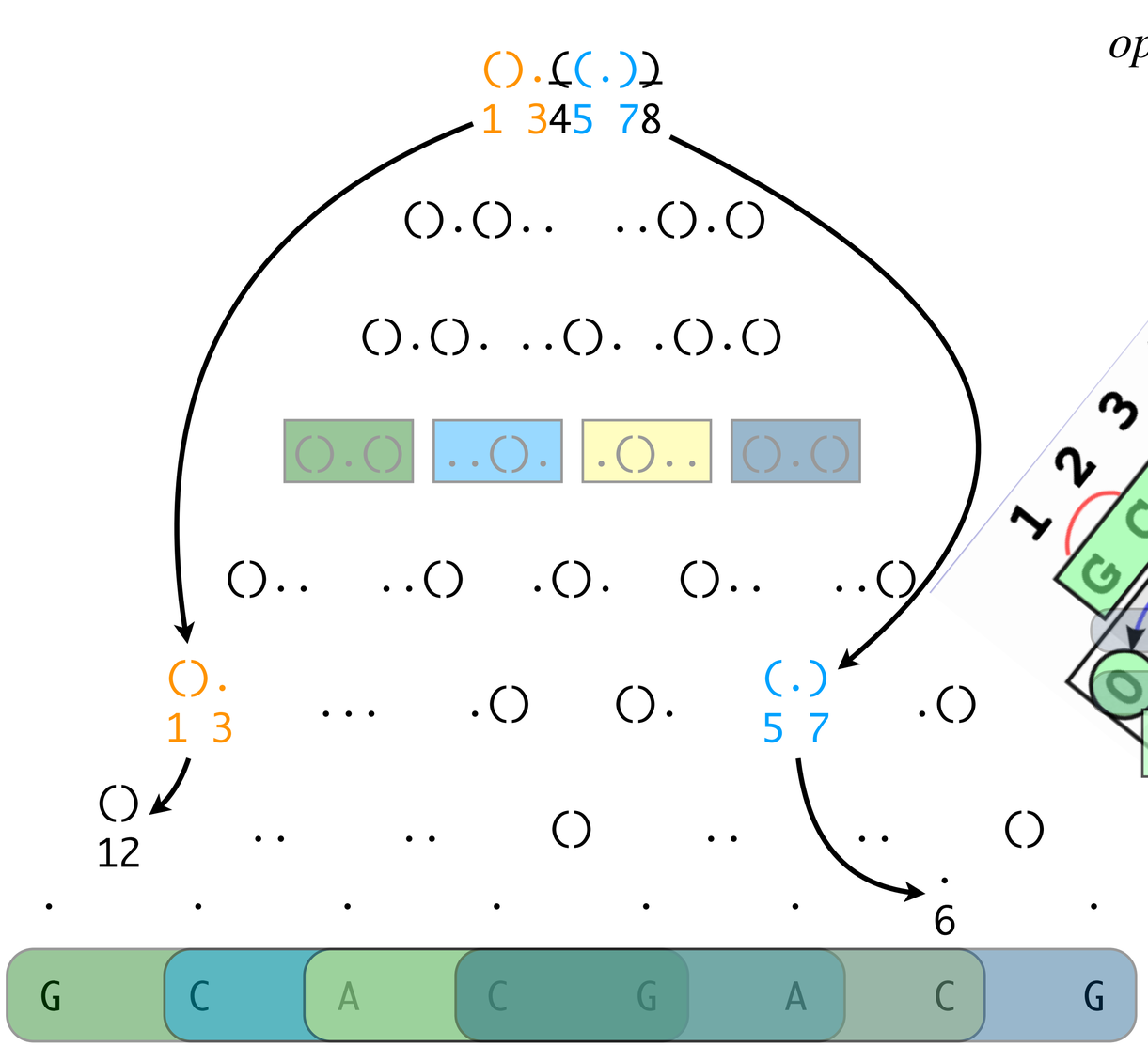
12345678
GCACGACG
xxx(xxx)
GCA
xx.
GC
()
().
GAC
(x)
A
.
(.)
().((.))

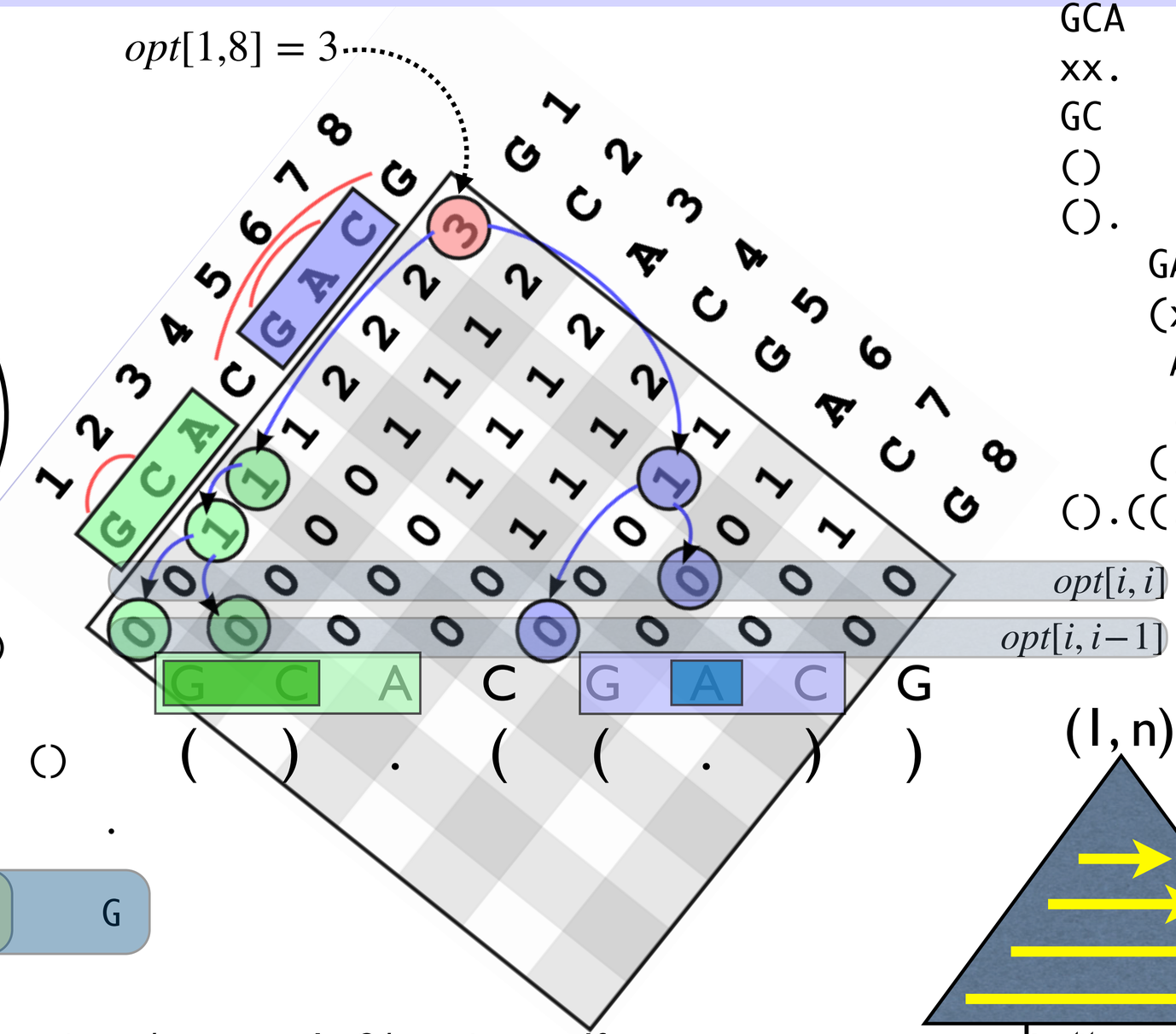
(I, n)

bottom-up

https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

RNA Folding Example: Nussinov






```
12345678
GCACGACG
xxx(xxx)
GCA
xx.
GC
()
().
    GAC
    (x)
     A
     .
    (.)
().((.))
```

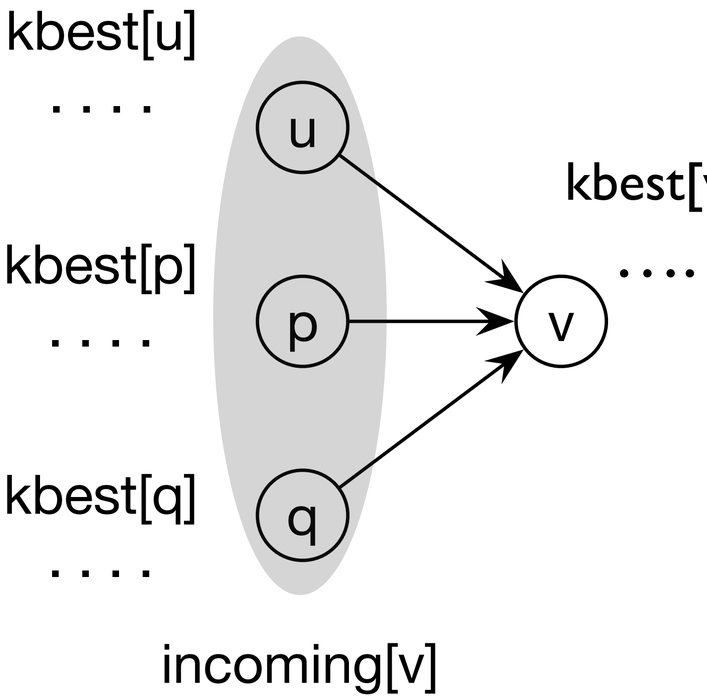
https://ad-publications.cs.uni-freiburg.de/student-projects/rna-google-2/nussinov.pdf

From 1-best to k-best

- each subproblem will now store top-k best answers instead of a single best
- we'll first extend Viterbi on DAGs to k-best Viterbi
- then extend generalized Viterbi on DAHs (e.g., CKY or Nussinov) to k-best

k-best Viterbi on Graph

- simple extension of Viterbi to solve k-best on graphs and hyper graphs
  cf. teams problem in HW4



for each node v,
  compute its kbest distances
  from the kbest of each incoming node u

1-best: $O(E+V)$

k-best: $O(E+V(d_{\max}+k\log d_{\max}))=O(E+E+Vk\log d_{\max})$
$=O(E+Vk\log d_{\max})$ where $d_{\max}$ is the max in-degree

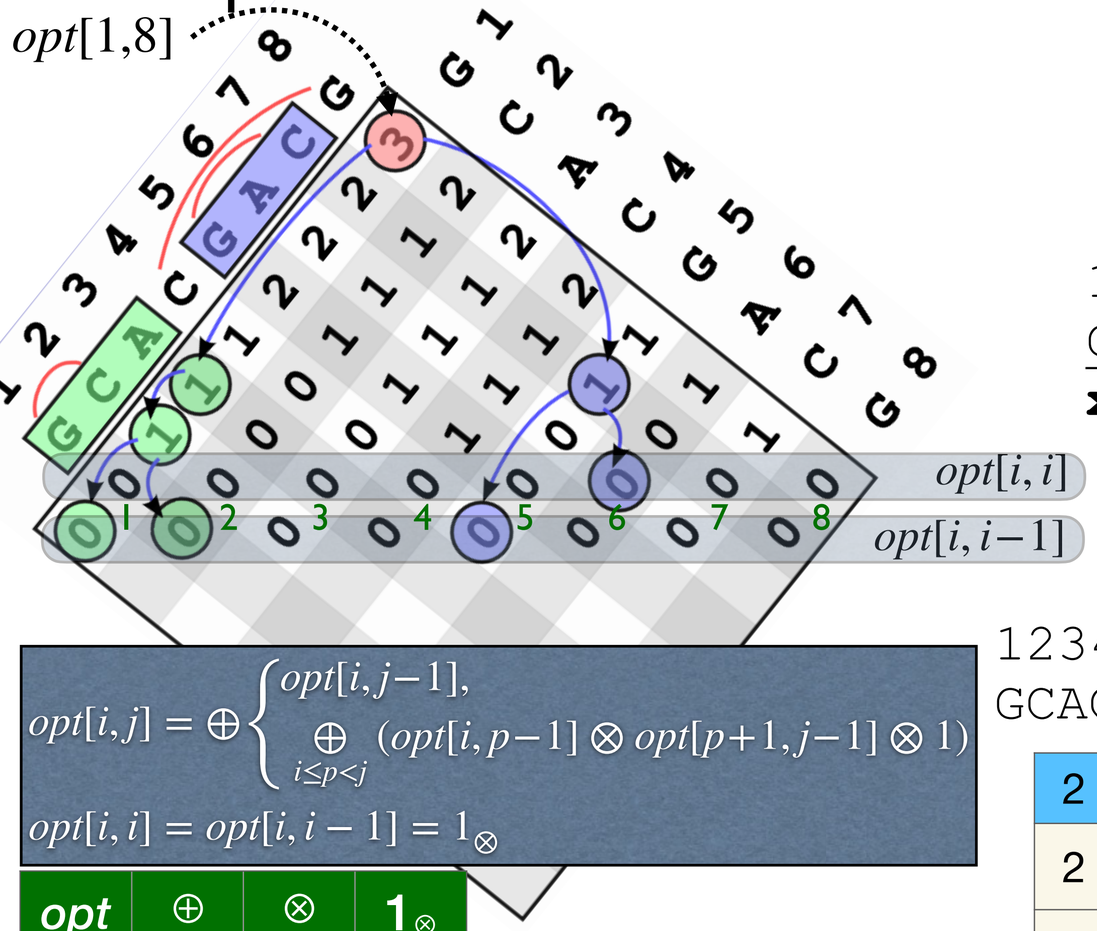
can improve it to: (cf. midterm & teams, w/ quickselect)

k-best: $O(E+Vk\log k)$ (assume $k \ll d_{\max}$)

("most states do not have anybody on team USA")

k-best Viterbi on Hypergraph

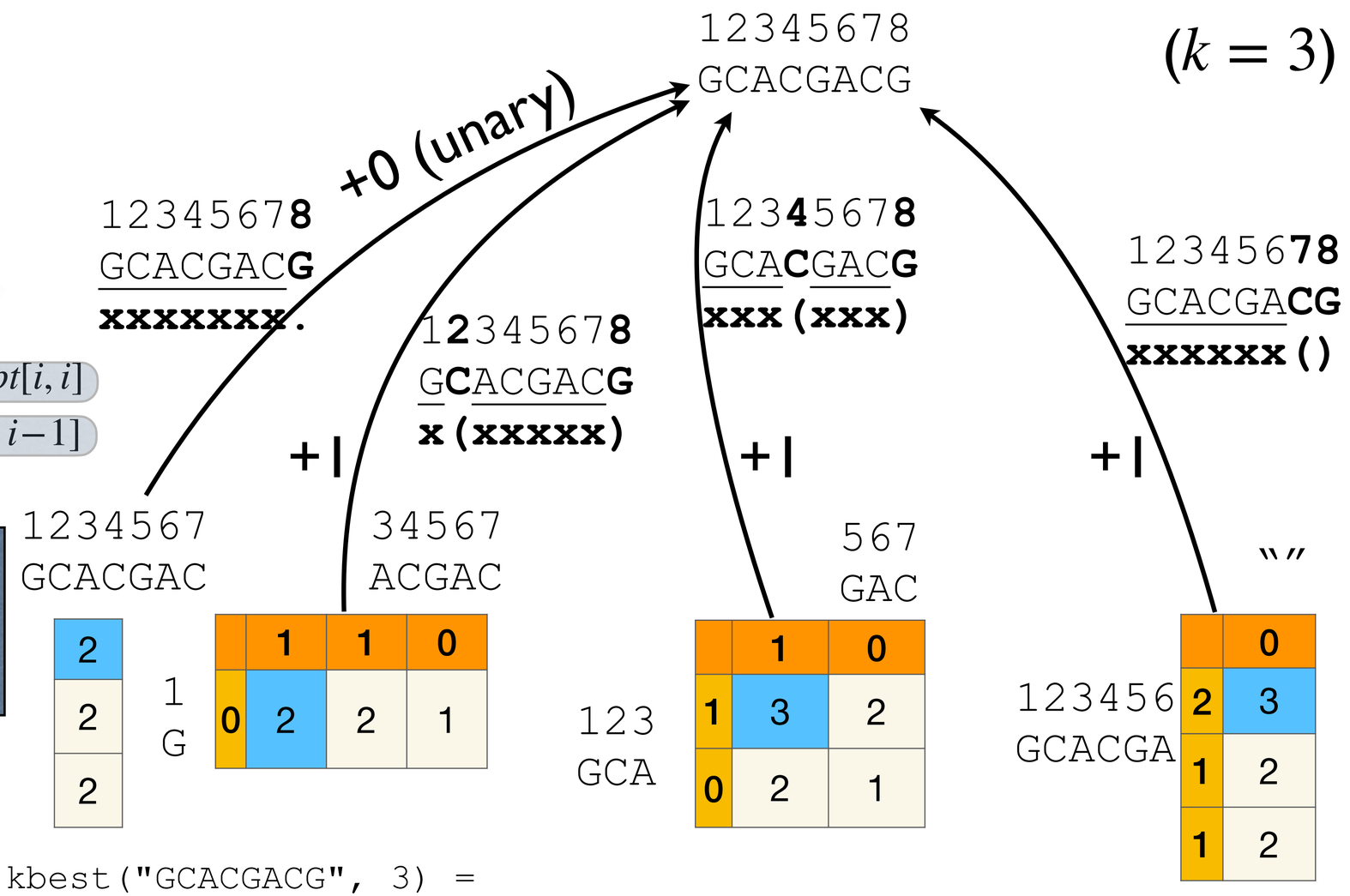
- simple extension of Viterbi to solve k-best on graphs and hyper graphs cf. midterm



$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

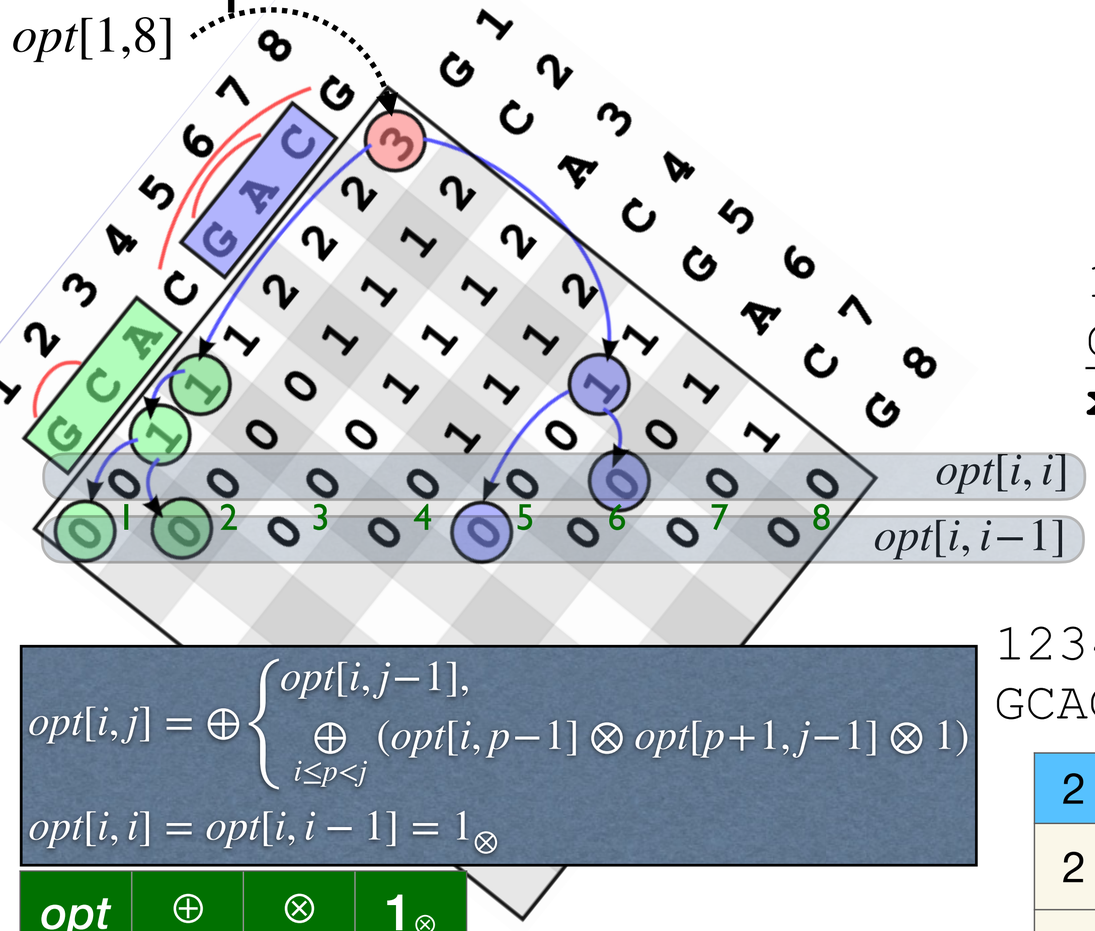
opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1



$(k = 3)$

k-best Viterbi on Hypergraph

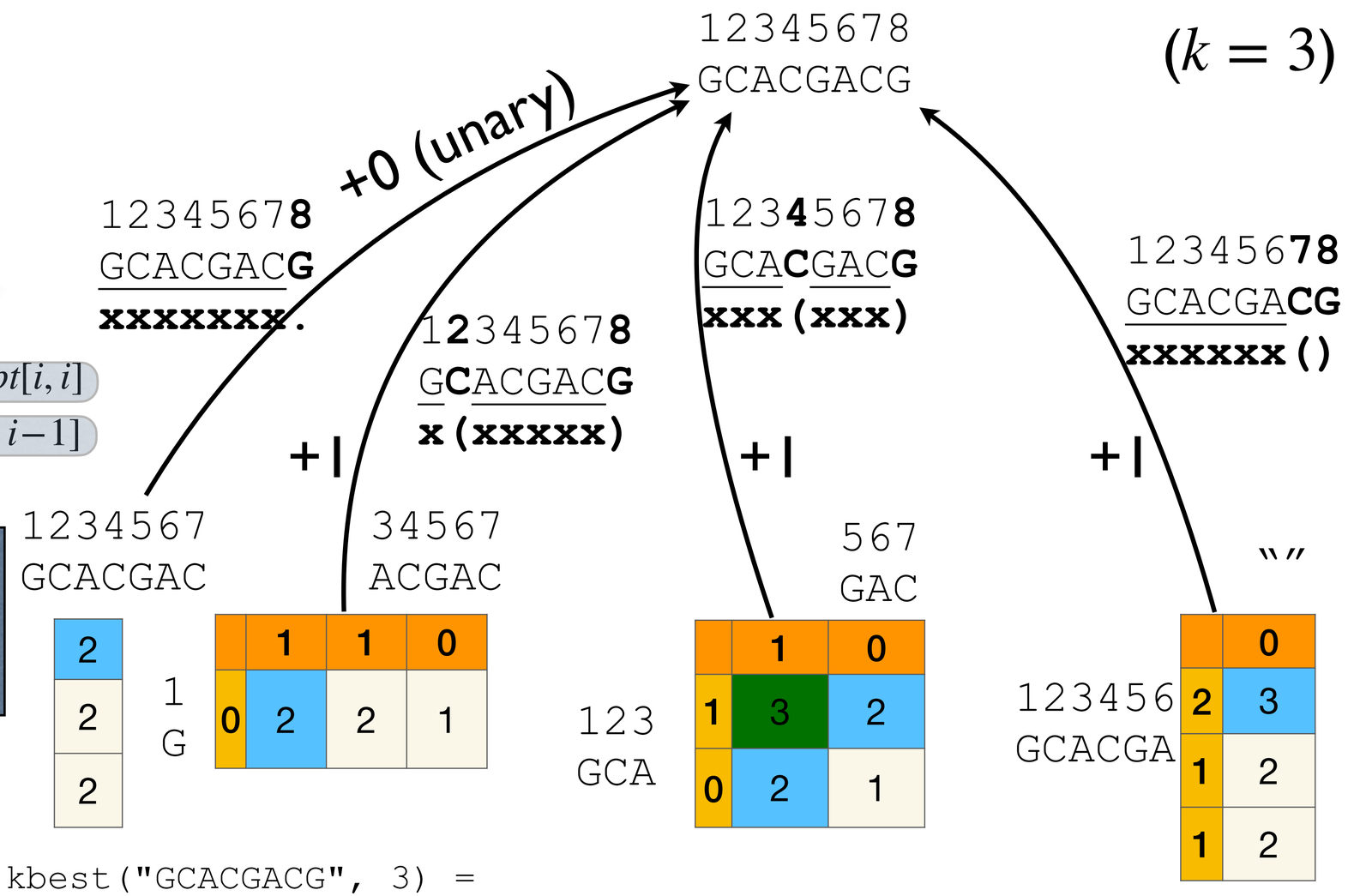
- simple extension of Viterbi to solve k-best on graphs and hyper graphs cf. midterm



$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

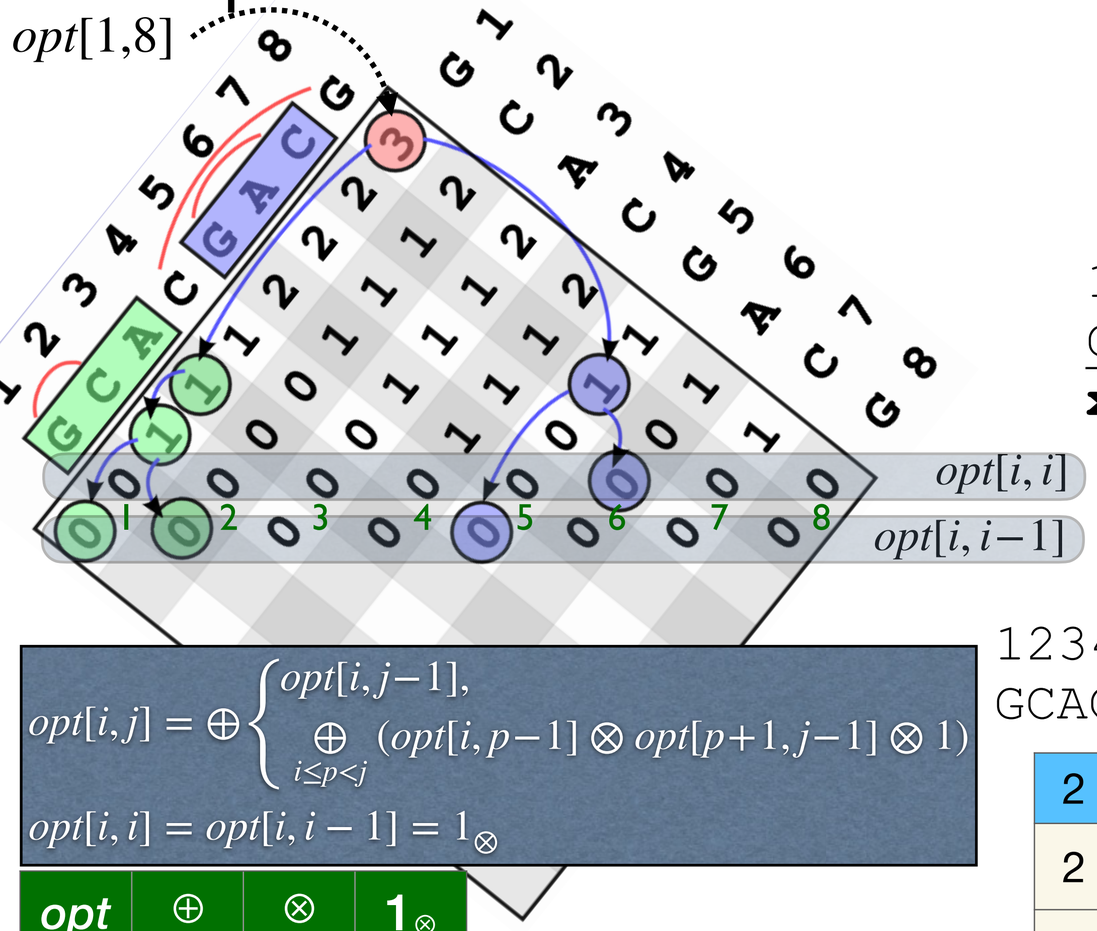
$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1



k-best Viterbi on Hypergraph

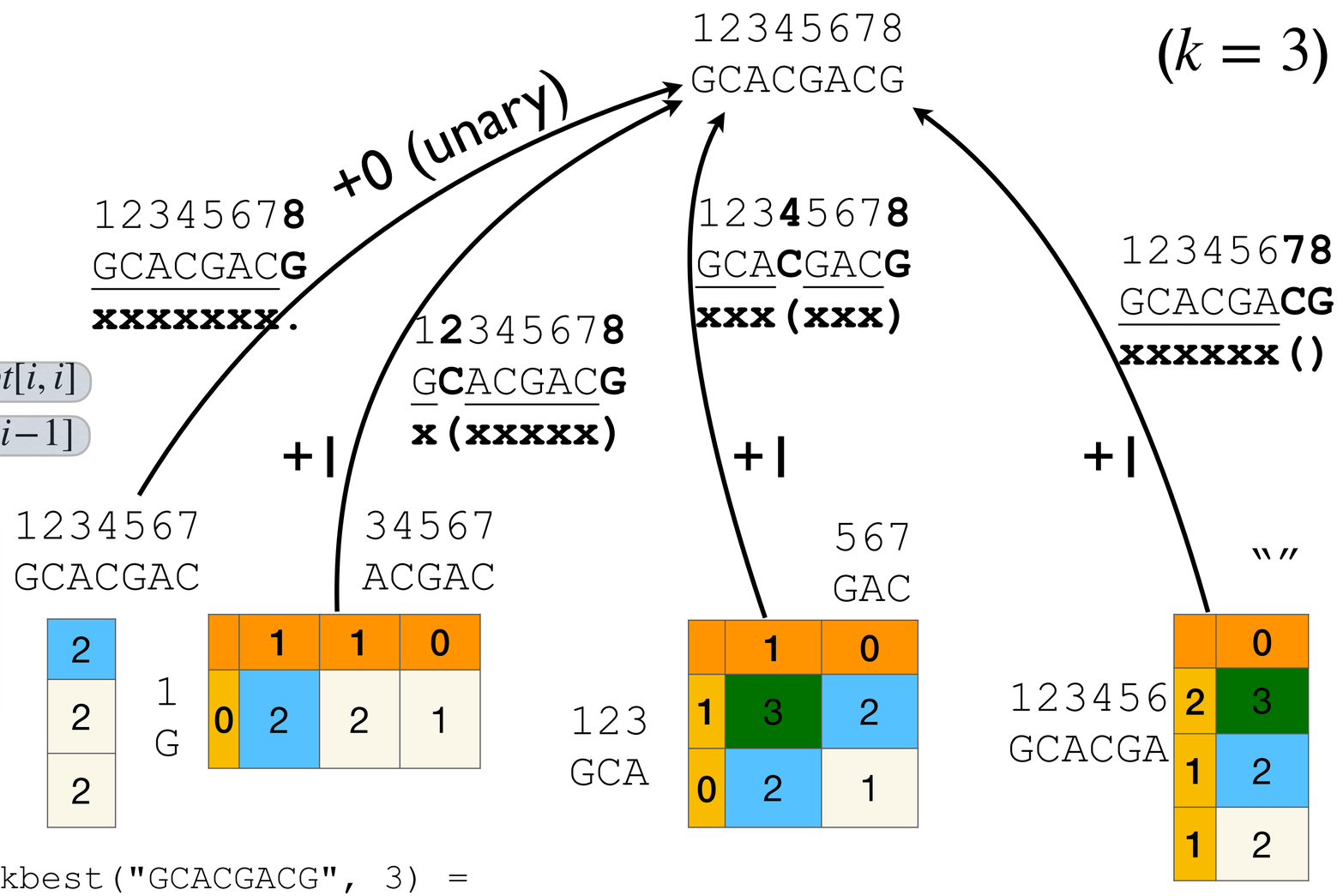
- simple extension of Viterbi to solve k-best on graphs and hyper graphs cf. midterm



$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

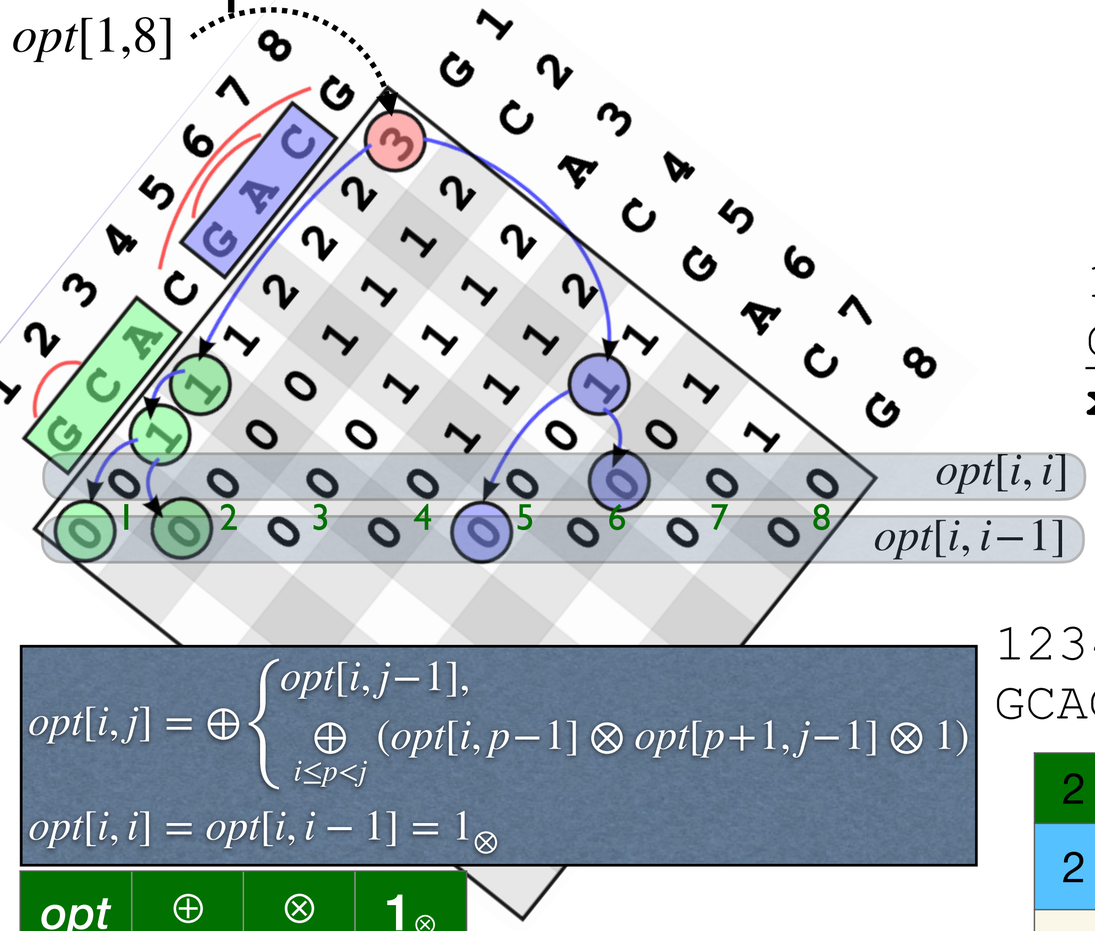
opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1




```
kbest("GCACGACG", 3) =
[(3, '().((.))'), (3, '().().()')]
```

k-best Viterbi on Hypergraph

- simple extension of Viterbi to solve k-best on graphs and hyper graphs cf. midterm

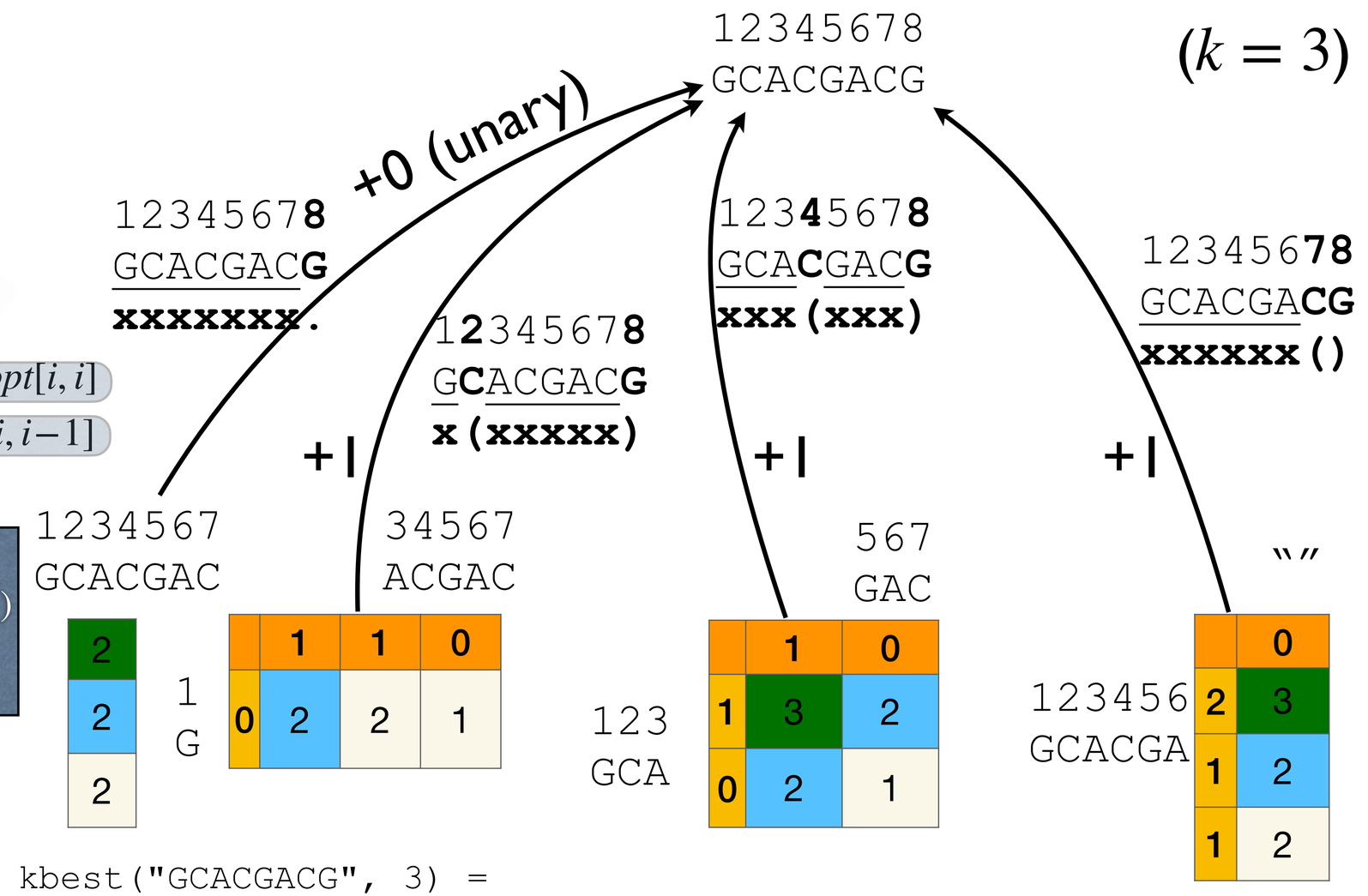


$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1

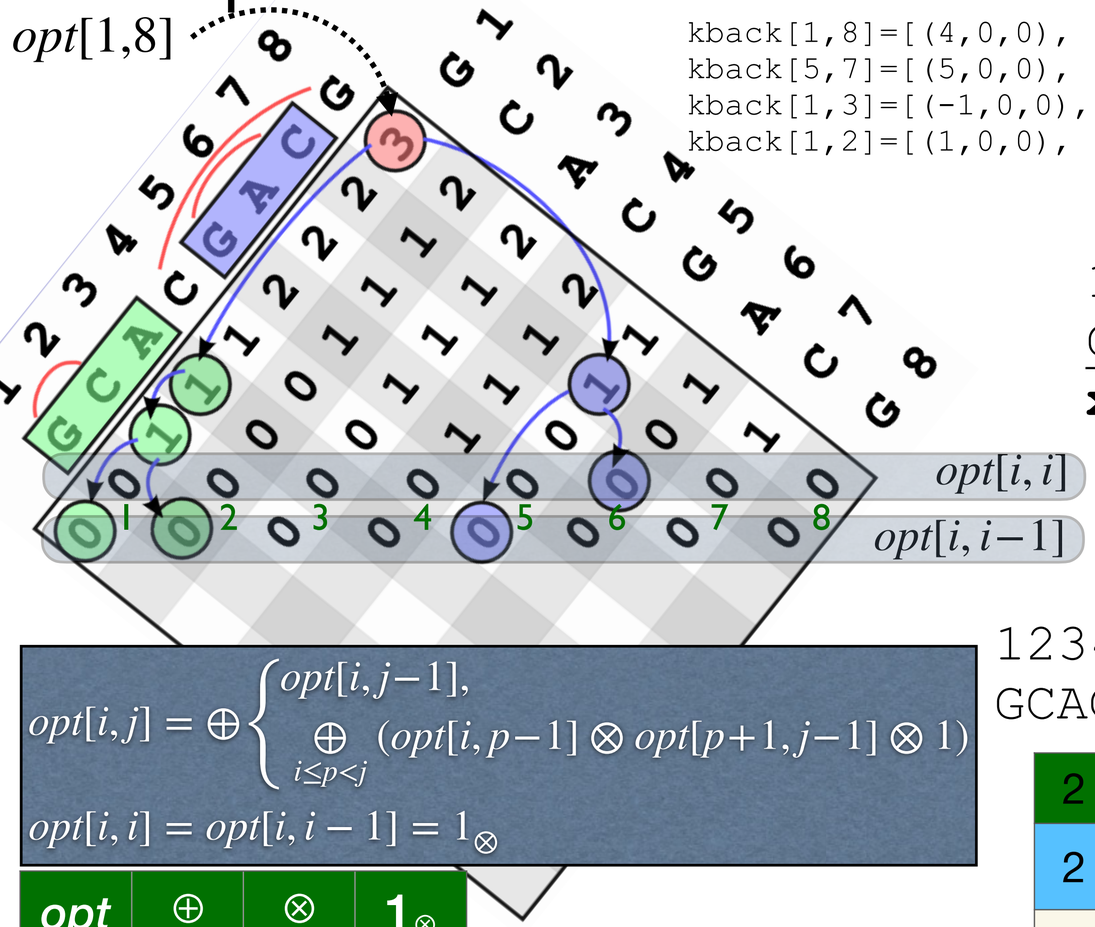
$(k = 3)$




```
kbest("GCACGACG", 3) =
[(3, '().((.))'), (3, '().().()'), (2, '().()._')]
```

k-best Viterbi on Hypergraph

- simple extension of Viterbi to solve k-best on graphs and hyper graphs cf. midterm




```
kback[1,8]=[(4,0,0), (7,0,0), (-1,0,0), (4,0,1)]
kback[5,7]=[(5,0,0), (-1,0,0)]
kback[1,3]=[(-1,0,0), (-1,1,0)]
kback[1,2]=[(1,0,0), (-1,0,0)]
```

$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

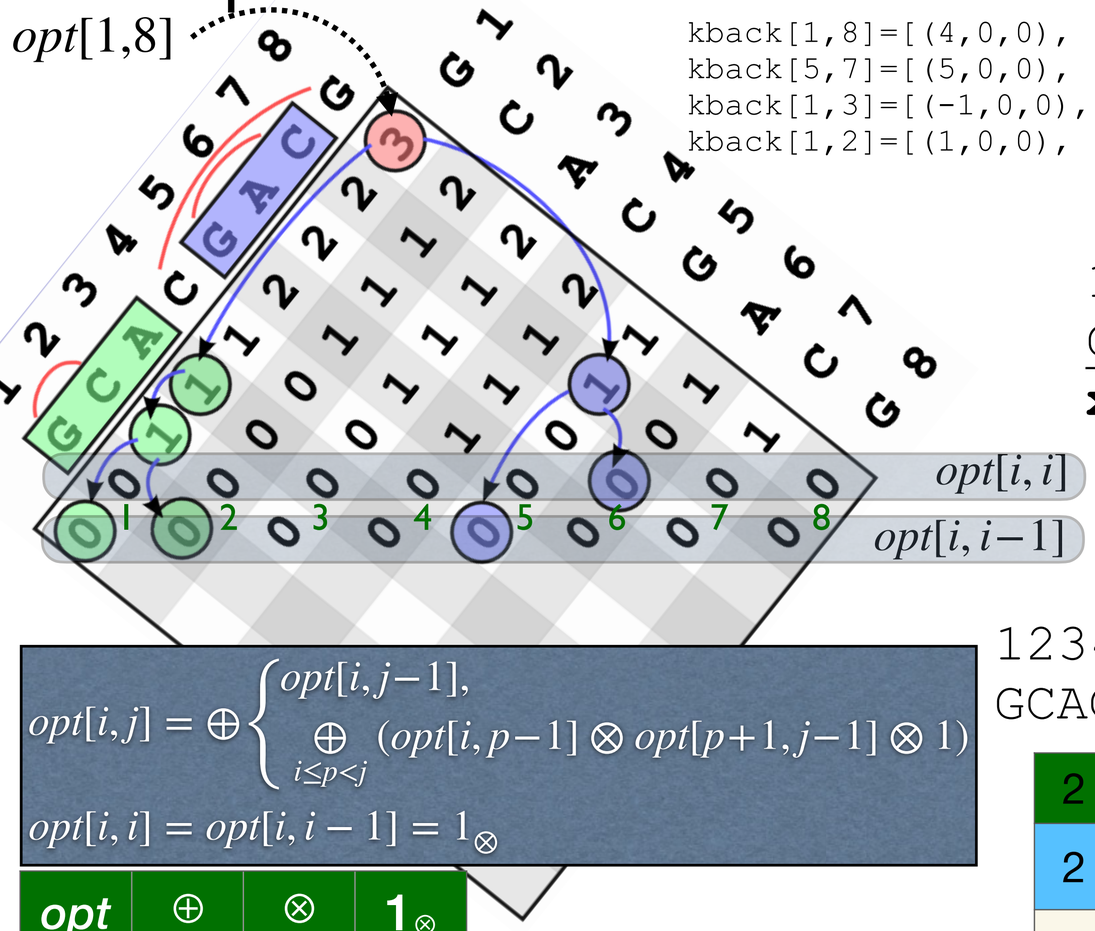
opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1


```
kbest("GCACGACG", 5) =
[(3, '().((.))'), (3, '().().()'), (2, '().()...'), (2, '().(...)')]
```

k-best Viterbi on Hypergraph

```
12345678 backtrace:
GCACGACG sol(1,8,4)
xxx(xxx)    [4,1,0]
GCA      sol(1,3,1)
xx.        [-1,1,0]
GC       sol(1,2,1)
x.         [-1,0,0]
G        sol(1,1,0)
.
    GAC  sol(5,7,0)
    (x)     [5,0,0]
     A   sol(6,6,0)
     .
...((.)) <-5th-best
```

- simple extension of Viterbi to solve k-best on graphs and hyper graphs

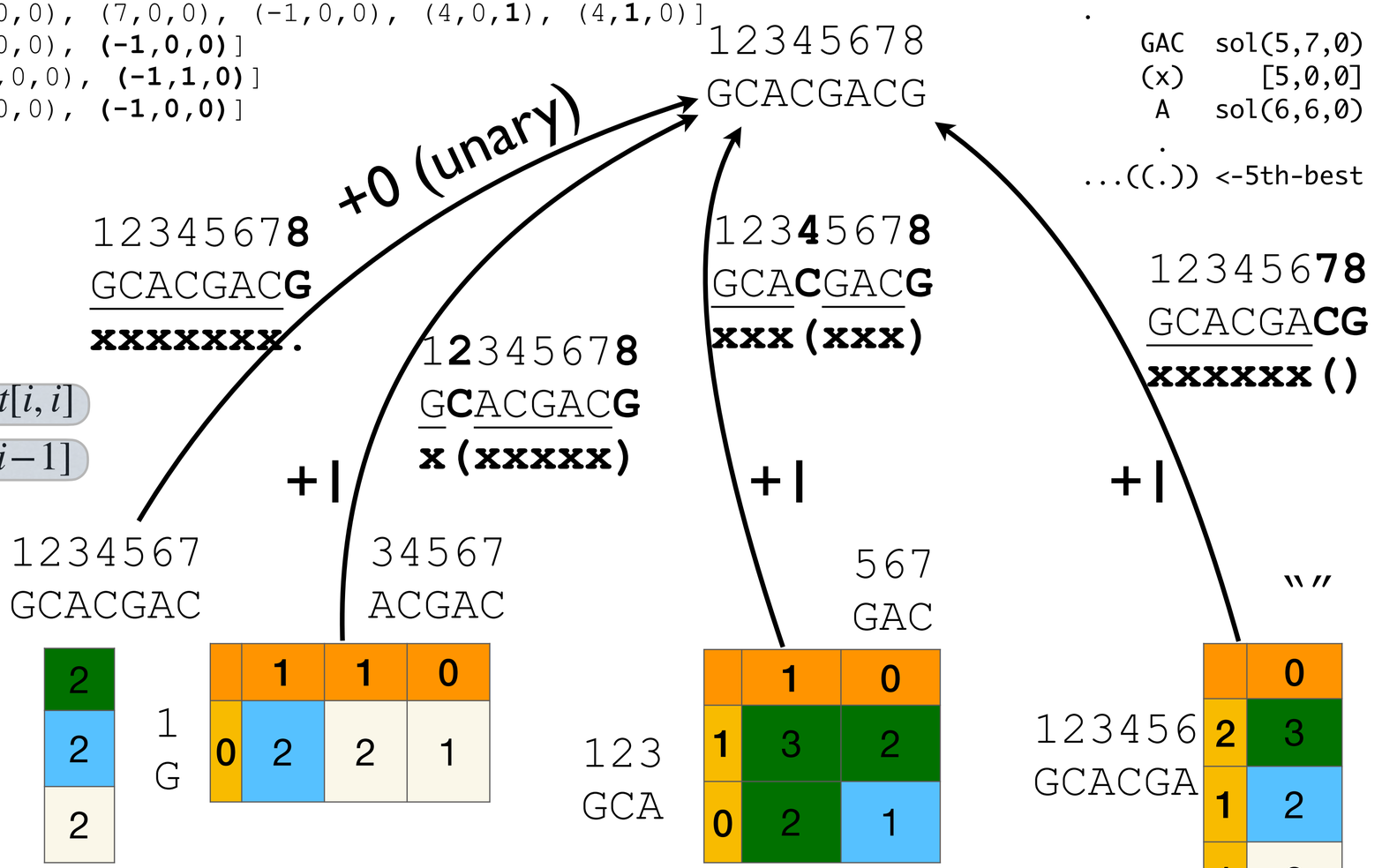



```
kback[1,8]=[(4,0,0), (7,0,0), (-1,0,0), (4,0,1), (4,1,0)]
kback[5,7]=[(5,0,0), (-1,0,0)]
kback[1,3]=[(-1,0,0), (-1,1,0)]
kback[1,2]=[(1,0,0), (-1,0,0)]
```

$$opt[i,j] = \oplus \begin{cases} opt[i,j-1], \\ \bigoplus_{i \le p < j} (opt[i,p-1] \otimes opt[p+1,j-1] \otimes 1) \end{cases}$$

$$opt[i,i] = opt[i,i-1] = 1_{\otimes}$$

opt	⊕	⊗	$1_{\otimes}$
best	max	+	0
total	+	x	1




```
kbest("GCACGACG", 5) =
[(3, '().((.))'), (3, '().().()'), (2, '().()...'), (2, '().(...)'), (2, '...((.))')]
```