CSE 566 Spring 2023

Four Russians' Algorithm for Edit Distance

Instructor: Mingfu Shao

Rap of the Edit Distance Problem

- **Edit Distance:** the minimum number of substitutions, insertions, and deletions that transforms X into Y.
- **Algorithms** (assume $m = |X|, n = |Y|$, and $n \geq m$)
    - Dynamic programming: $O(mn) = O(n^2)$
    - Wavefront: $O(nd)$, where d is the edit distance $d = \Theta(n)$
- **Bound**: if SETH is true, then the edit distance problem cannot be solved in $O(n^{2-\delta})$ time for any $\delta > 0$.
- **Four Russian's Algorithm**: $O(n^2/\log n)$.

Partitioning

$|X| = |Y| = n$

- Partition the DP table into blocks. $k \times k$
- Adjacent blocks overlaps one row/ column.
- Size of block: k, to be determined.

$k = 3$

	A	C	C	G	T	G	C
A	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
G	2	1	1	1	2	3	4
G	3	2	2	1	2	2	3
A	4	3	3	2	2	3	3
T	5	4	4	3	2	3	4
C	6	5	4	4	3	3	3

Partitioning

- Partition the DP table into blocks.
- Adjacent blocks overlaps one row/ column.
- Size of block: k, to be determined.

	A	C	C	G	T	G	C
A	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
G	2	1	1	1	2	3	4
G	3	2	2	1	2	2	3
A	4	3	3	2	2	3	3
T	5	4	4	3	2	3	4
C	6	5	4	4	3	3	3

A Single Block b

K

Y_b

X_b

K

	A	C	C	G	T	G
A	1	2	3	3	4	3
C	2					**4**
G	2					**3**
G	3					**4**
A	2					**3**
T	1	**2**	**3**	**4**	**4**	**3**

K^2

$$f(X_b, Y_b, L_b, T_b) \rightarrow (R_b, B_b)$$

Framework

$\# \text{blocks} = \left(\frac{n}{K}\right)^2$

```
1. preprocessing for f functions
2. init the first row and column
3. DP in the unit of blocks
   for i = 1 to n/k
       for j = 1 to n/k
           call f on block b
           (indexed by i and j)
       end for
   end for
4. return bottom-right number
```

	A	C	C	G	T	G	C
A	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
G	2	1	1	1	2	3	4
G	3	2	2	1	2	2	3
A	4	3	3	2	2	3	3
T	5	4	4	3	2	3	4
C	6	5	4	4	3	3	3

(2,3)

trivial: $O\left(\left(\frac{n}{K}\right)^2 \cdot K^2\right) = O(n^2)$

$\max\{i, j\}$

Preprocessing

$DP[i, j] \in \underline{\{0, 1, \cdots, n\}}$

- Compute and store (R_b, B_b) for every (X_b, Y_b, L_b, T_b).
- How many possible inputs?

$|\Sigma|^{k+k-1} \cdot (n+1)^{2k-1}$

	A	C	C	G	T	G
A	1	2	3	3	4	3
C	2					**4**
G	2					**3**
G	3					**4**
A	2					**3**
T	1	**2**	**3**	**4**	**4**	**3**

$\rightarrow Y_b$

$\rightarrow T_b$

$\rightarrow R_b$

$f(X_b, \underline{Y_b}, L_b, T_b) \rightarrow (R_b, B_b)$

Offset Encoding #1

	A	C	C	G	T	G
A	2	3	4	4	5	4
C	3					**5**
G	3					**4**
G	4					**5**
A	3					**4**
T	2	**3**	**4**	**5**	**5**	**4**

$\xrightarrow{-2}$

	A	C	C	G	T	G
A	$\boxed{0}$	1	2	2	3	2
C	1					**3**
G	1					**2**
G	2					**3**
A	1					**2**
T	0	**1**	**2**	**3**	**3**	**2**

- $f(X_b, Y_b, L_b + C, T_b + C) \rightarrow (R_b + C, B_b + C)$
- We only compute/score inputs where top-left number is 0.

Offset Encoding #2

- **Fact:** in the DP table, adjacent values differ by at most 1.
- Encode a number with {0, +1, -1}.

	A	C	C	G	T	G	
A	0	1	2	2	3	2	→ Tb
C	1						
G	1						
G	2						
A	1						
T	0						
	↓ Lb						

	A	C	C	G	T	G
A	0	+	+	0	+	−
C	+					
G	0					
G	~~0~~ +					
A	~~+~~ 0 −					
T	−					

Preprocessing Revisited

- Compute and store (R_b, B_b) for every (X_b, Y_b, L_b, T_b).
- How many inputs now?

$|\Sigma|^{2K-1} \cdot 3^{2K-2}$

$= \frac{1}{|\Sigma|} \cdot \frac{1}{9} \cdot (3|\Sigma|)^{2K}$

	A	C	C	G	T	G
A	0	+	+	0	+	−
C	+					
G	0					
G	0					
A	+					
T	−					

$O(K^2)$

$f(X_b, Y_b, L_b, T_b) \rightarrow (R_b, B_b)$

Preprocessing Revisited

log n

- Let $k = (\log_{3|\Sigma|} n)/2$.
- #possible-input:

$O\left((3|\Sigma|)^{2k}\right) = O(n)$

- Solve each case with DP: $O(k^2)$
- Total running time of preprocessing:

$O(nk^2) = O(n \cdot \log^2 n)$

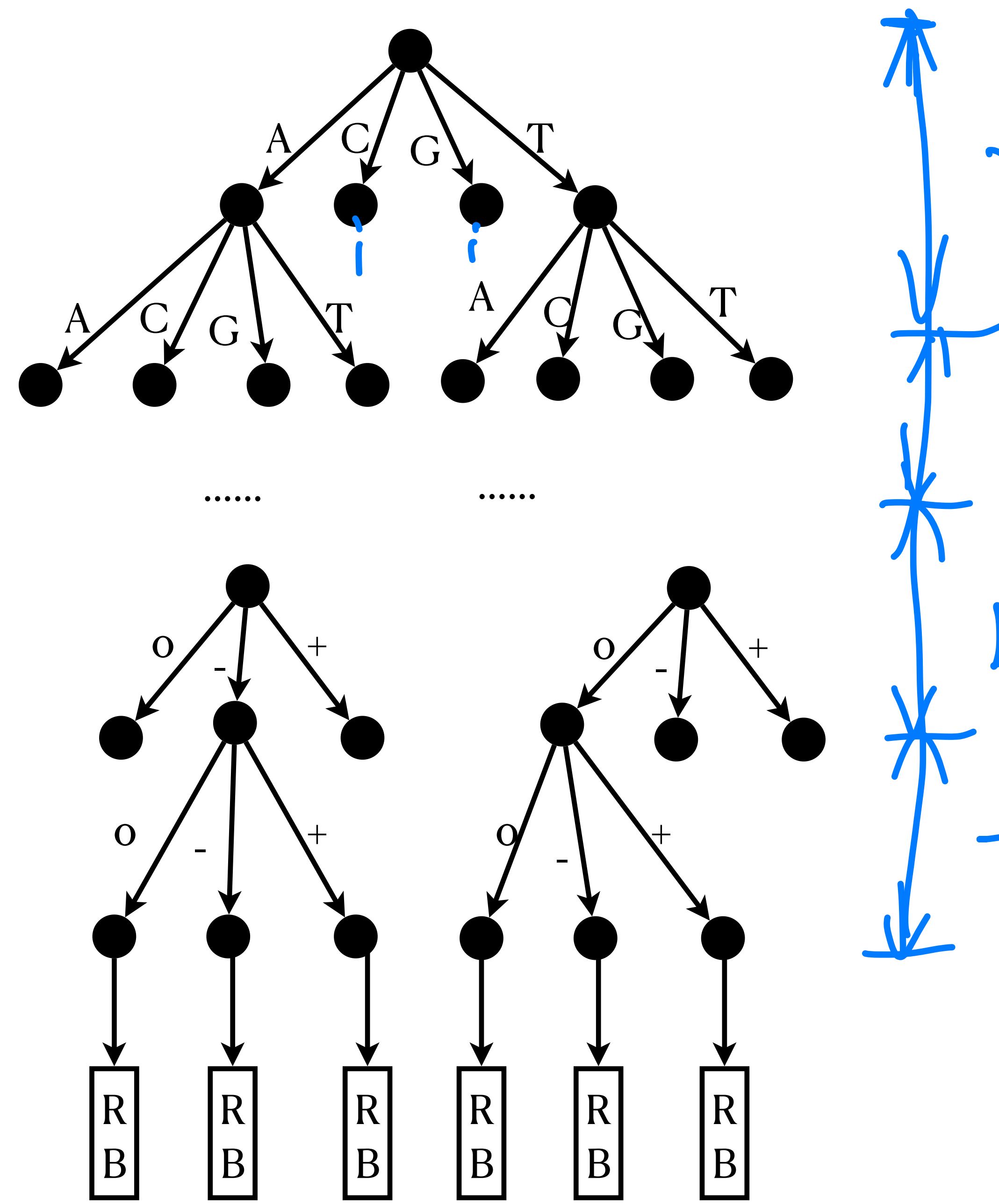
	A	C	C	G	T	G
A	0	+	+	0	+	−
C	+					
G	0					
G	0					
A	+					
T	−					

$f(X_b, Y_b, L_b, T_b) \rightarrow (R_b, B_b)$

Storing and Querying f

- Store with a tree/trie.
- Each path encodes a possible input/output.
- Space: $O(n)$
- Query: $O(k) = O(\log n)$

$4k = O(\log n)$



Analysis

```
1. preprocessing for f functions
2. init the first row and column
3. DP in the unit of blocks
   for i = 1 to n/k
       for j = 1 to n/k
         call f on block b
         (indexed by i and j)
       end for
   end for
4. return bottom-right number
```

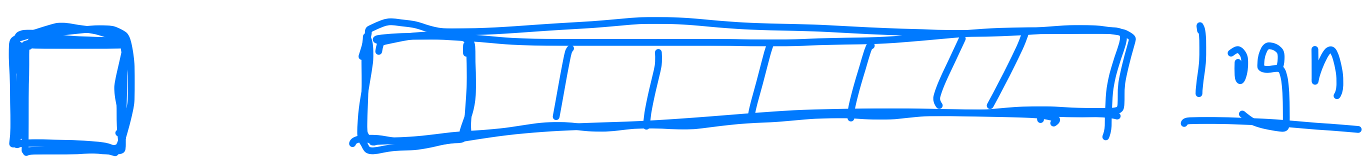
$\rightarrow O(n \cdot \log^2 n)$

$\rightarrow O(n)$

$O\left(\frac{n^2}{k^2} \cdot k\right) = O\left(\frac{n^2}{\log n}\right)$

Summary

- Why it works?!
- Method of Four-Russian
  - Idea: build a look-up table of logarithmic size
  - Other applications: transitive closure of graphs, multiplication of boolean matrix
  - Speed up by a factor of $\log n$ or $\log^2 n$
- Key: one can afford enumeration of logarithmic size.



$\{0,1\}$

$2 \cdot 2 \cdots \cdots 2 \cdot 2 = 2^{\log n} = O(n)$