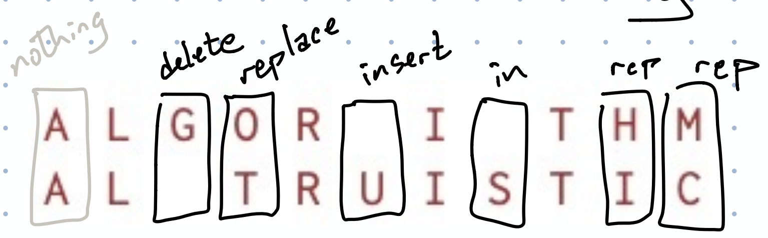
HW0 due 9pm

HW1 due in one week — groups of ≤3

Office hours

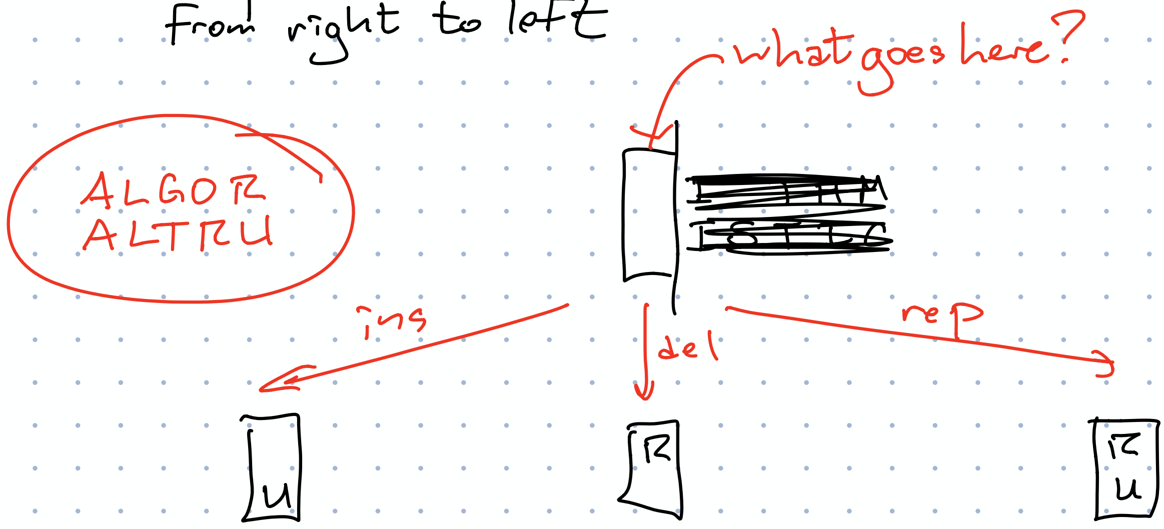
ClassTranscribe

Edit Distance: Minimum # insertions, deletions, and replacements required to change one string into another.



ALGORITHM
↓
ALORITHM
↓
ALTRITHM
↓
ALTRUITHM
↓
⋮

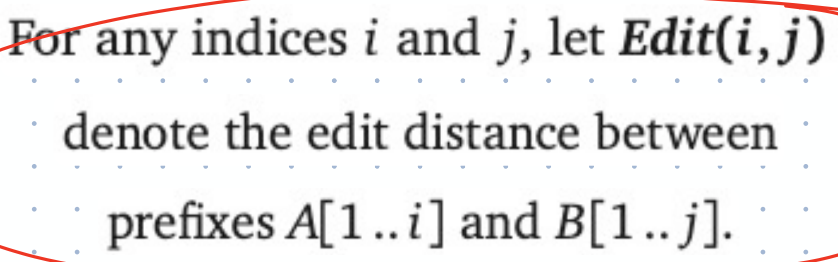
Build the optimal motif from right to left



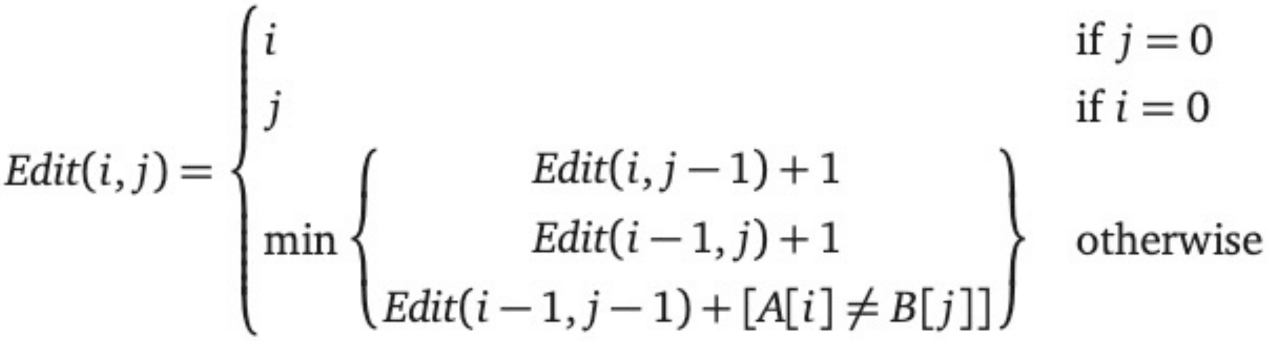
Find the edit distance between any two prefixes.

$$\boxed{\text{Edit}(i,j) = \text{edit distance between } A[1..i] \text{ and } B[1..j]}$$

general recursive case: $\text{Edit}(i,j) = \min \begin{cases} 1 + \text{Edit}(i,j-1) \\ 1 + \text{Edit}(i-1,j) \\ [A[i] \neq B[j]] + \text{Edit}(i-1,j-1) \end{cases}$

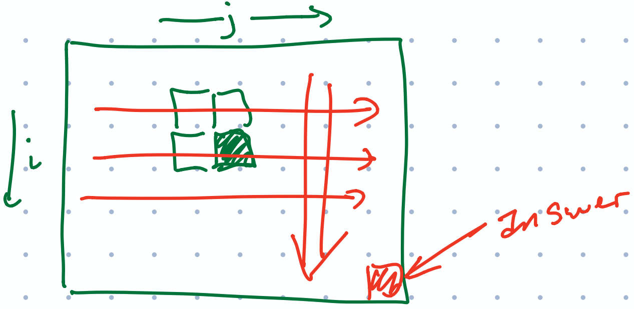


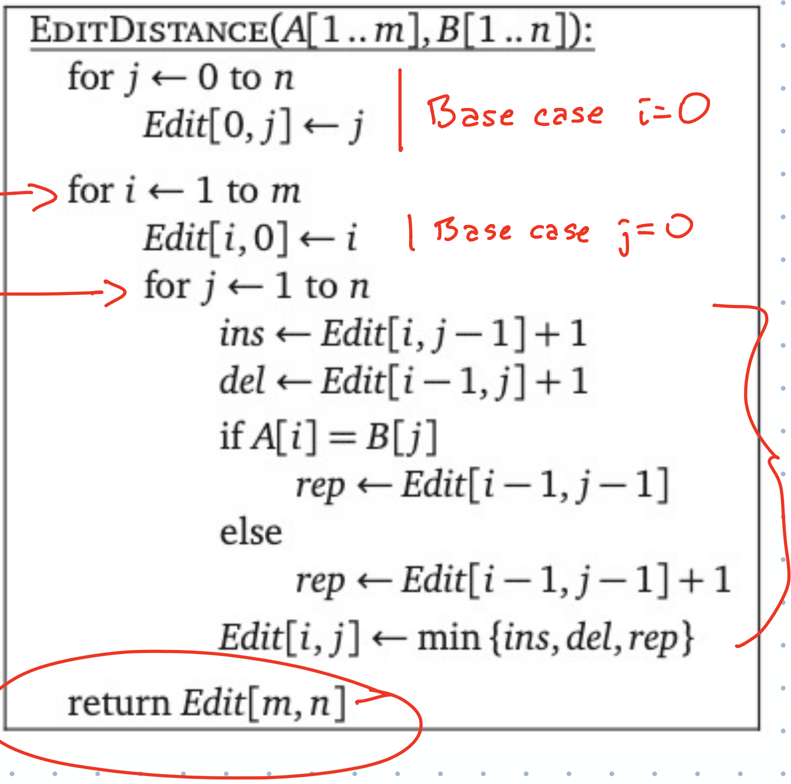
For any indices i and j, let ***Edit*(*i*, *j*)** denote the edit distance between prefixes $A[1..i]$ and $B[1..j]$.



$$Edit(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \left\{ \begin{array}{c} Edit(i, j-1) + 1 \\ Edit(i-1, j) + 1 \\ Edit(i-1, j-1) + [A[i] \neq B[j]] \end{array} \right\} & \text{otherwise} \end{cases}$$

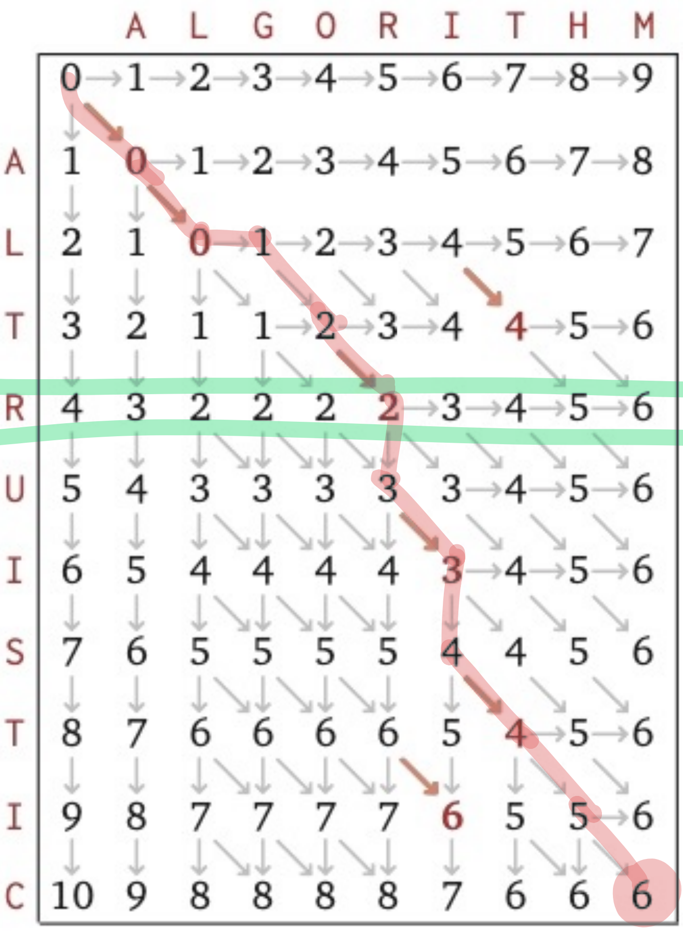
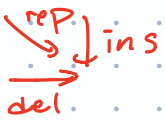
Input: $A[1..m]$ and $B[1..n]$ ⇒ we need $Edit(m,n)$





```
EditDistance($A[1..m], B[1..n]$):
    for $j \leftarrow 0$ to $n$                         | Base case i=0
        $Edit[0, j] \leftarrow j$
    for $i \leftarrow 1$ to $m$
        $Edit[i, 0] \leftarrow i$                      | Base case j=0
        for $j \leftarrow 1$ to $n$
            $ins \leftarrow Edit[i, j-1] + 1$
            $del \leftarrow Edit[i-1, j] + 1$
            if $A[i] = B[j]$
                $rep \leftarrow Edit[i-1, j-1]$
            else                                       recurrence
                $rep \leftarrow Edit[i-1, j-1] + 1$
            $Edit[i, j] \leftarrow \min\{ins, del, rep\}$
    return $Edit[m, n]$
```

$O(mn)$ time



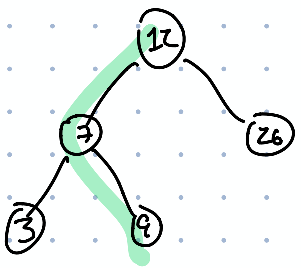
		A	L	G	O	R	I	T	H	M
	0	1	2	3	4	5	6	7	8	9
A	1	0	1	2	3	4	5	6	7	8
L	2	1	0	1	2	3	4	5	6	7
T	3	2	1	1	2	3	4	4	5	6
R	4	3	2	2	2	2	3	4	5	6
U	5	4	3	3	3	3	3	4	5	6
I	6	5	4	4	4	4	3	4	5	6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	4	5	6
I	9	8	7	7	7	7	6	5	5	6
C	10	9	8	8	8	8	7	6	6	6

```
A L G O R I   T H M
A L T R U I S T I C

A L G O R   I   T H M
A L   T R U I S T I C

A L G O R   I   T H M
A L T   R U I S T I C
```

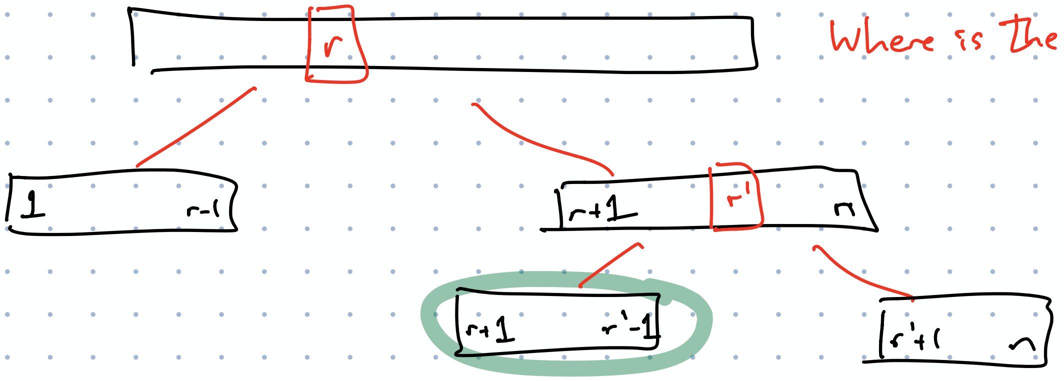
Optimal Binary Search Trees



Given search frequencies $f[1..n]$ of n keys,

build the BST T that minimizes

$$\sum_i f[i] \cdot cost(T, i)$$



Where is the root?

Subproblems: intervals $f[i..j]$

$OptCost(i,j)$ = total cost of all searches in best BST for keys $i..j$ with frequencies $F[i..j]$

$$\mathrm{Cost}(T, 1, n) = \sum_{i} F(i) \cdot \mathrm{Cost}(T, i)$$

$$\mathrm{cost}(T, i) = \begin{cases} 1 & i = \mathrm{root} \\ 1 + \mathrm{cost}(\mathrm{left}(T), i) & i < \mathrm{root} \\ 1 + \mathrm{cost}(\mathrm{right}(T), i) & i > \mathrm{root} \end{cases}$$

$$\mathrm{Cost}(T, 1, n) = \sum_{i=1}^{n} F[i] + \mathrm{Cost}(\mathrm{left}(T), 1, \mathrm{root}-1) + \mathrm{Cost}(\mathrm{right}(T), \mathrm{root}+1, n)$$

$$\mathrm{OptCost}(i, k) = \sum_{j=i}^{k} F[j] + \min_{i \le j \le k} \left\{ \begin{matrix} \mathrm{OptCost}(i, j-1) \\ + \\ \mathrm{OptCost}(j+1, k) \end{matrix} \right\}$$