Suffix Arrays

1 2 3 4 5 6 7 8

A B A B A A B A

(8 5 6 3 1 7 4 2)

n Suffixes

$O(n \log n) \times O(n)$

Cost of comparison

naive

$O(n^2 \log n)$

we can get $O(n)$ time by using suffix trees!

$O(n)$ time direct methods exist.

q: BA

naive querying

$O(|q| \log n)$

↓

$O(|q| + \log n)$

1	2	3	4	5	6	7	8
A	B	A	B	A	A	B	A

1	2	3	4	5	6	7	8
8	5	6	3	1	7	4	2

L_1 M_1 M_2 R

L_1

$q = BA$

$l = LCP(q, L) = 0$

Longest common prefix length

$r = LCP(q, R) = 2$

1) maintain mlr

↳ start any comparison at $mlr + 1$ (Position)

$mlr = 0$

$mlr = \min(l, r)$

we find BA.

2) start at max of l, r

L →	10	A B A
	15	A B A A C
	2	A B A B C
M →	1	A B A C
	3	A B C
	5	A B C A
R →	20	A B C D

$q = ABC$

$l = 2 = LCP(q, L)$

$r = 3 = LCP(q, R)$

$mlr + 1 = 2 + 1 = 3$

$O(|q| \cdot \log n)$

1	2	3	4	5	6	7	8
A	B	A	B	A	A	B	A

1	2	3	4	5	6	7	8
8	5	6	3	1	7	4	2

L, M, R

$q = BA$

$Lcp(q, L) \quad l = 0$

$Lcp(q, R) \quad r = 2$

$Lcp(L, M) = 0$

$Lcp(M, R) = 2$

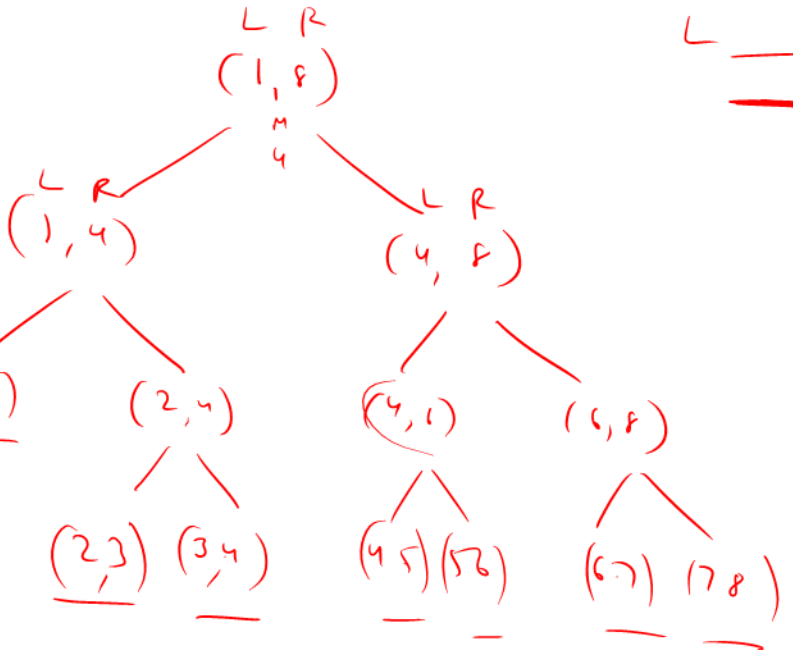
Precomputed Value

q matches suffix 7

$O(|q| + \log n)$

look only "once"

binary search



n − 1 internal nodes

7 leaves

n leaves

$O(n + n - 1) = O(n) \times O(1) = O(n)$

Inexact Matching via Hashing

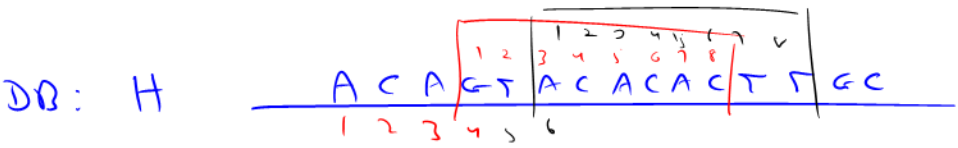
l ← read length

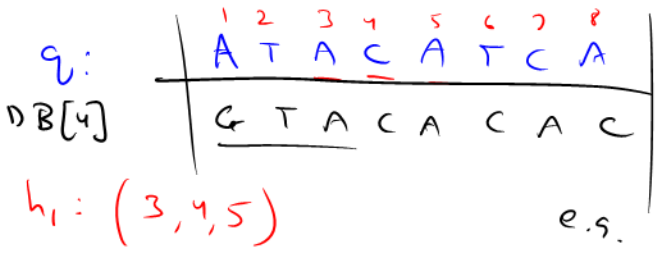
d ← # of mismatches allowed

LSH : Locality sensitive Hashing

h_1 : Choose any k positions at random

→ hash key



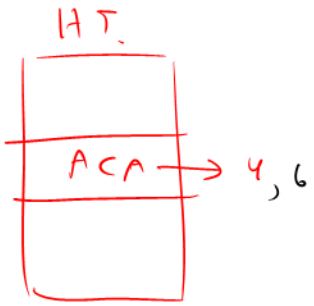


h_1 : (3, 4, 5) e.g. $h(1,2,3)$

$l = 8$

$d = 4$

$k = 3$



Prob of a match between q & DB for a single position

Answer : DB(4)
DB(6)

$$\left(1 - \frac{d}{l}\right)$$

Prob of a match over k positions is $\left(1 - \frac{d}{l}\right)^k$

1) false positive

→ db substring hashes to a cell (DB[i]), but the true distance > d

✓

solution: just compare the q with DB[i]

2) false negative:

there exists a match, but they don't hash to the same cell?

solution: we cannot eliminate this

however, we can minimize this!

do m hash keys

$h_1(1,2,3)$ ⟶ $P(\text{of mismatch}) \left(1-\left(1-\frac{d}{d}\right)^k\right)$

$h_2(7,4,5)$

$h_3(\ \ \ )$

$\vdots$

$h_m(\ \ \ )$

Prob of a false negative (mismatch) even after m hash functions

$$\left(1-\left(1-\frac{d}{d}\right)^k\right)^m < \varepsilon$$

ε is the desired prob of a mismatch.

e.g. $\varepsilon = 0.001$

solve for m!

The Tree of Life : Evolutionary Tree

Phenotype { morphological or physical characteristics

↓

genotype (complete gene, genome

molecular basis

DNA / Protein

fine grained

Inversions
splitting
merging

coarser-level rearranging the genes

---

1) Parsimony-based methods

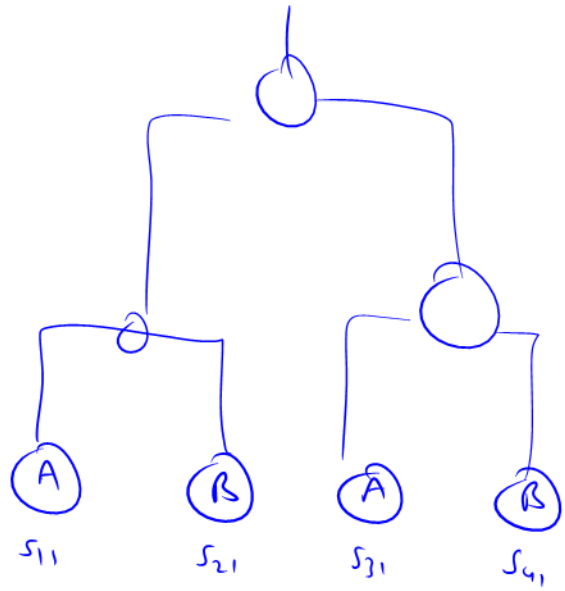
most economical tree from # of changes

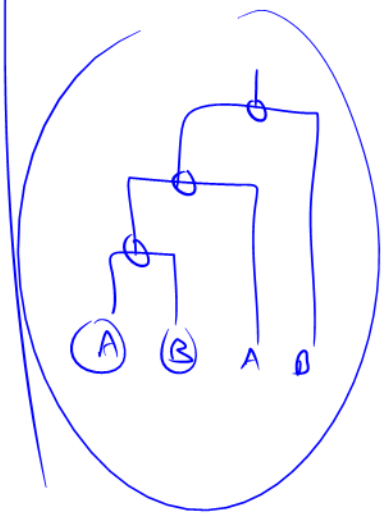
assumption: all positions are independent

⇒ add up the cost across all positions

# of changes

s_1 :	A	B	A
s_2 :	B	A	A
s_3 :	A	A	B
s_4 :	B	B	A





Given: search over binary trees

1) search over different topologies

$(2n-3)!!$ rooted trees

$\equiv (2n-1)(2n-5)(2n-7)\cdots$ (1)

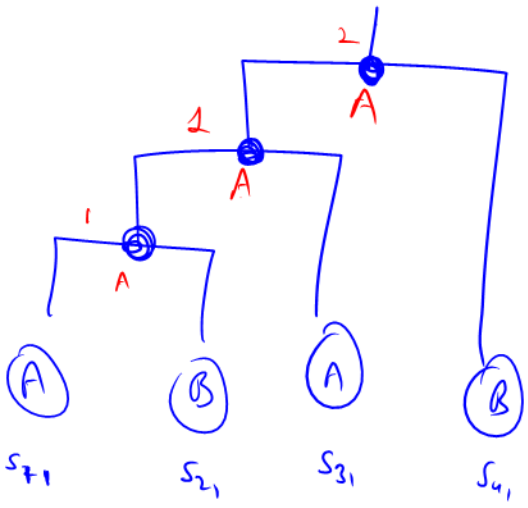
infeasible to enumerate all trees

→ sampling

e.g. MCMC

Markov Chain Monte Carlo

2) how to compute the cost of a given tree



Upper-bound (3)

bottom up: node: If $L \cap R = \phi$

then $L \cup R$ is kept

else

keep $L \cap R$

top-down phase:

starting at the root, for any node

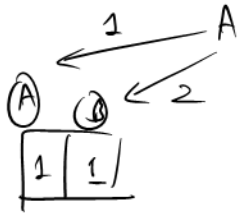
If more than one choice

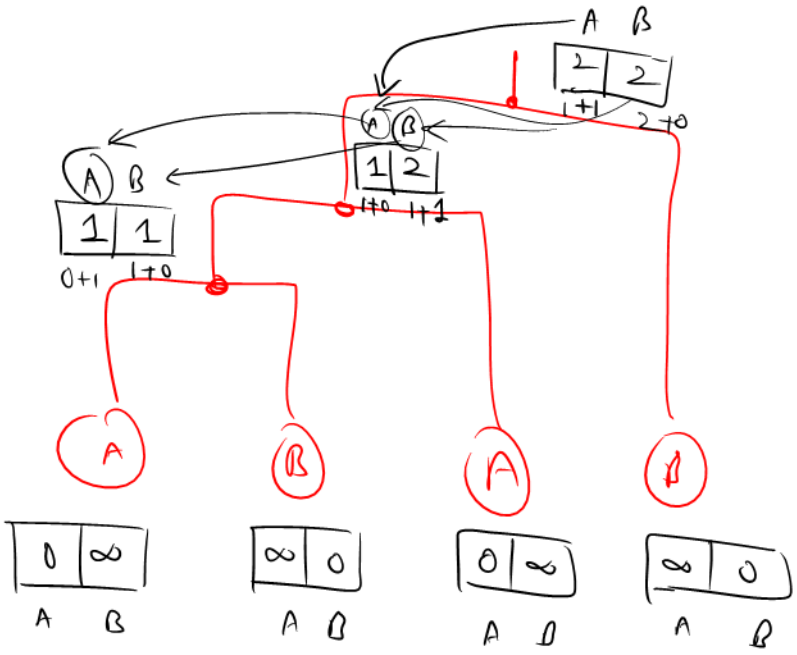
use parent's symbol

	A	B
A	0	1
B	1	0

Dynamic programming

find the least cost tree.





2 2
A B