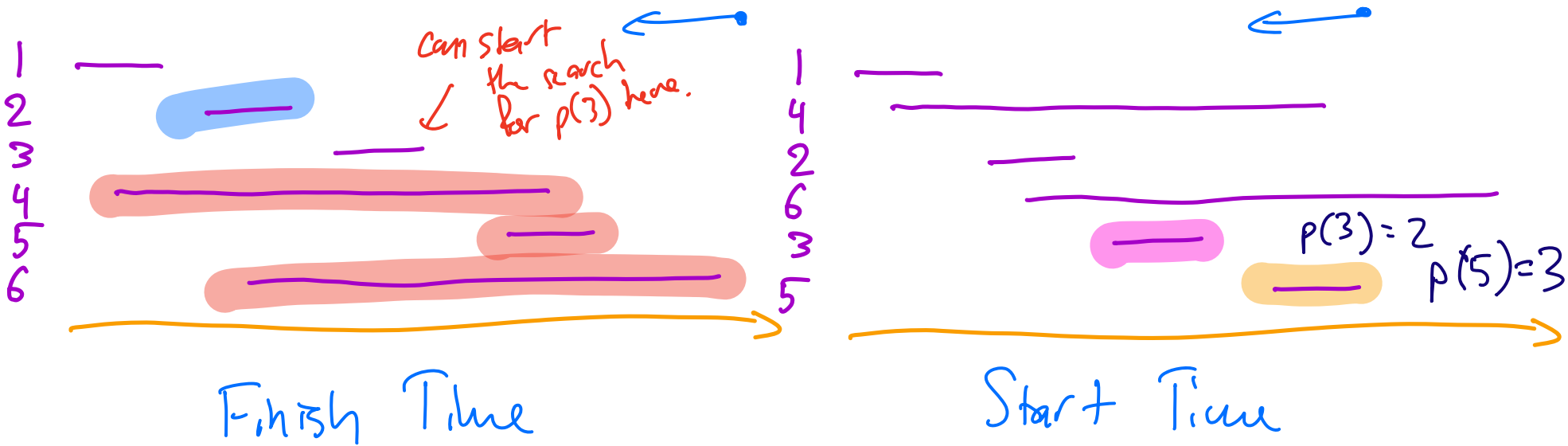
9/21

5112

Sequence Alignment

Randomization

Weighted Interval Scheduling



Sequence Alignment

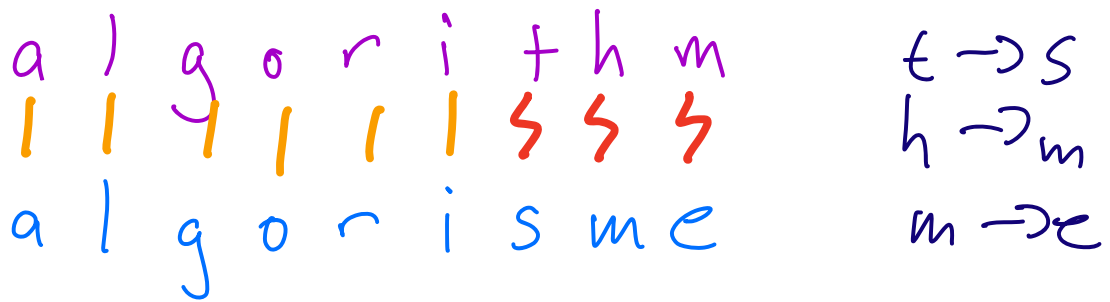
a l g o r i t h m English

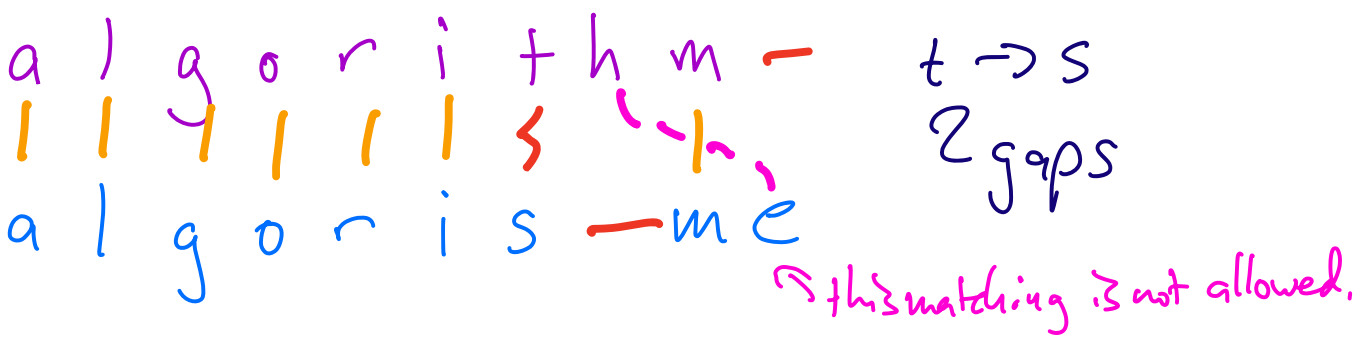
a l g o r i s m e Old French

a l g o r i s m u s Medieval Latin

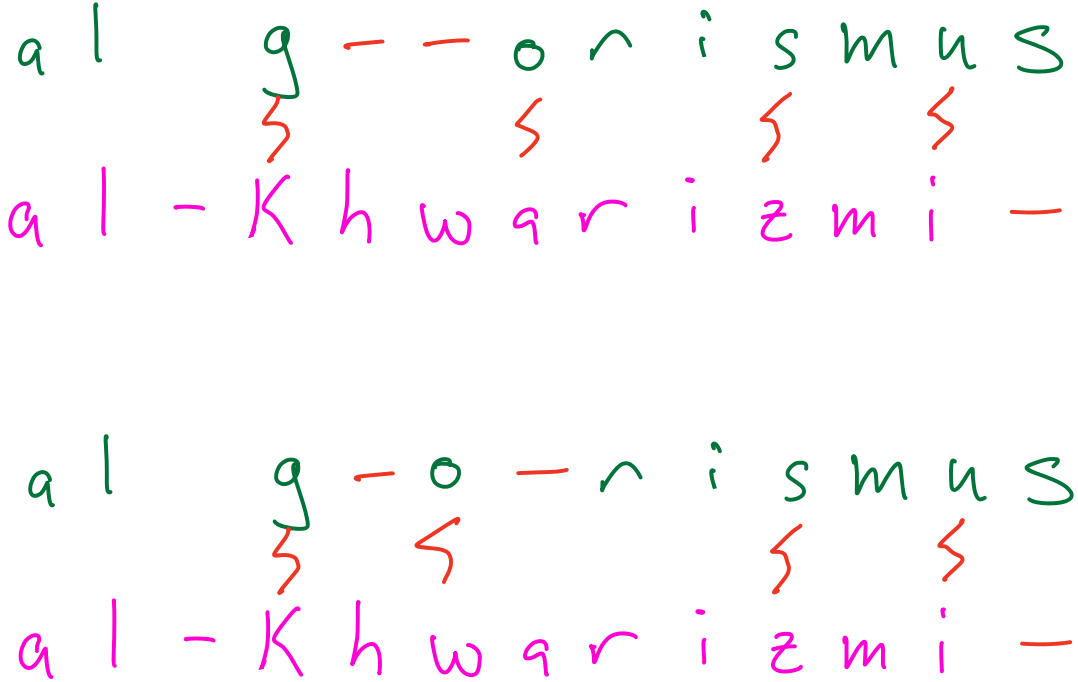
a l - K h w a r i z m i Arabic

Sequence Alignment





Sequence Alignment



Intuition:
Want different costs for different mismatches.

o → a small cost

o → w large cost

Sequence Alignment

Cost table: gap cost δ
mismatch cost $\alpha_{x,y}$ for each pair $x, y \in \Sigma$

Input: two strings X, Y for alphabet Σ

Output: Alignment which minimizes total cost

Set M of matched pairs (i, j) which each pair indicates the i^{th} pos. of X is matched to the j^{th} pos. of Y.

Condition:
$(i, j) \in M$ and $(k, l) \in M$
$\Rightarrow$ either $i \leq k$ and $j \leq l$
or $i \geq k$ and $j \geq l$.

Sequence Alignment

1 2 3 4 5 6 7 8 9 10

a l g - - o r i s m s

a l - K h w a r i z m i - —

1 2 3 4 5 6 7 8 9 10 11

M

(1,1) (2,2) (3,3) (4,6)
(5,7) (6,8) (7,9) (8,10)
(9,11)

Cost: $\alpha_{g,k} + \alpha_{o,a} + \alpha_{s,z} + \alpha_{u,i} + 35$

a l g - o - r i s m s

a l - K h w a r i z m i - —

Sequence Alignment

X a l g o r i t h m — n characters

Y a l g o r i s m u s — m characters

not allowed

Look at the last characters.

1. $(n, m) \in M$
2. The n^{th} pos. in X is unmatched
3. The m^{th} pos. in Y is unmatched.

Define $Opt(i, j)$ to be the value of the optimal alignment of the first i characters of X with the first j characters of Y.

Recursive cases:

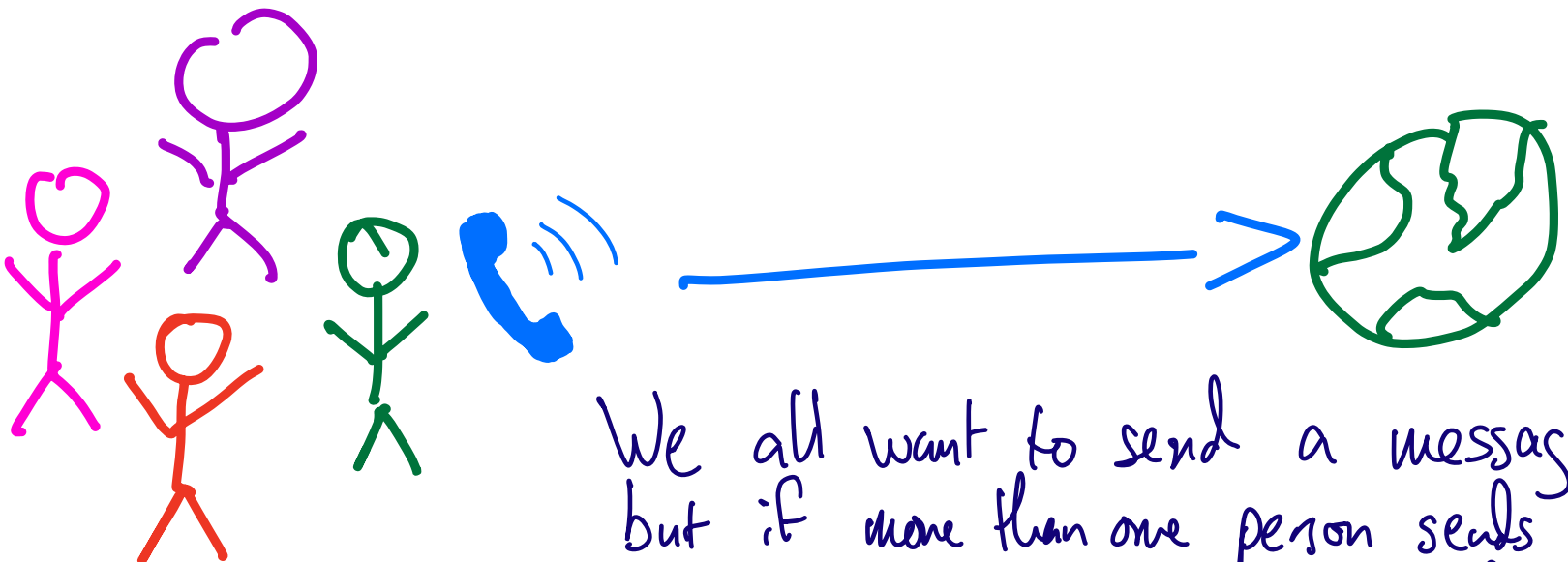
1. $(i, j) \in M$ $\quad \alpha_{x(i), y(j)} + Opt(i-1, j-1)$
2. X's n^{th} pos unmatched $\quad \delta + Opt(i-1, j)$
3. Y's m^{th} pos unmatched $\quad \delta + Opt(i, j-1)$

Randomization

Algorithms which use randomness

- Execution depends on randomness
- Always outputs correctly eventually
- Running time may depend on the randomness

Shared Channel



We all want to send a message, but if more than one person sends a message at a time, the messages don't send.

n clients

At each time t, a client can either try to send a message or not.

No feedback: can't tell if ~~your~~ any message sends

Each client has to execute the same algorithm
No communication!

Deterministic algorithms never send a single message.

A Randomized Algorithm

At each timestep send the message with probability p.

Define $A(i,t)$ to be the event that client i send a message at time t.

$\overline{A(i,t)}$, the complement of $A(i,t)$: the event where $A(i,t)$ didn't happen.

$Pr(A(i,t)) = p \quad \forall\, i \text{ and } t. \qquad Pr(\overline{A(i,t)}) = 1 - p$