Scalability and Distributed Systems

Bill Howe

INFX 575: Data Science III

Scaling, Applications, and Ethics

This week…

- Quiz
- Scalability
- MapReduce
- Class Exercise: Launching a Spark cluster

- Twitter Assignment due tonight
- Project Proposal due next week

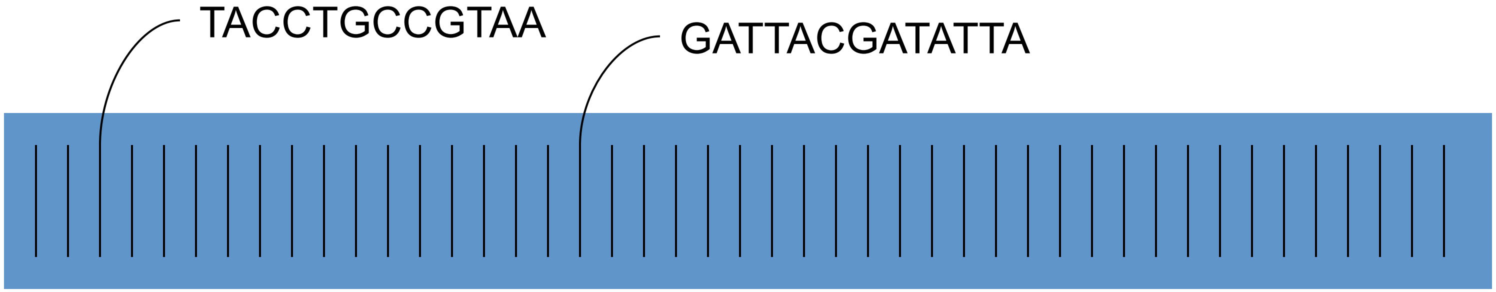
https://pollev.com/billhowe

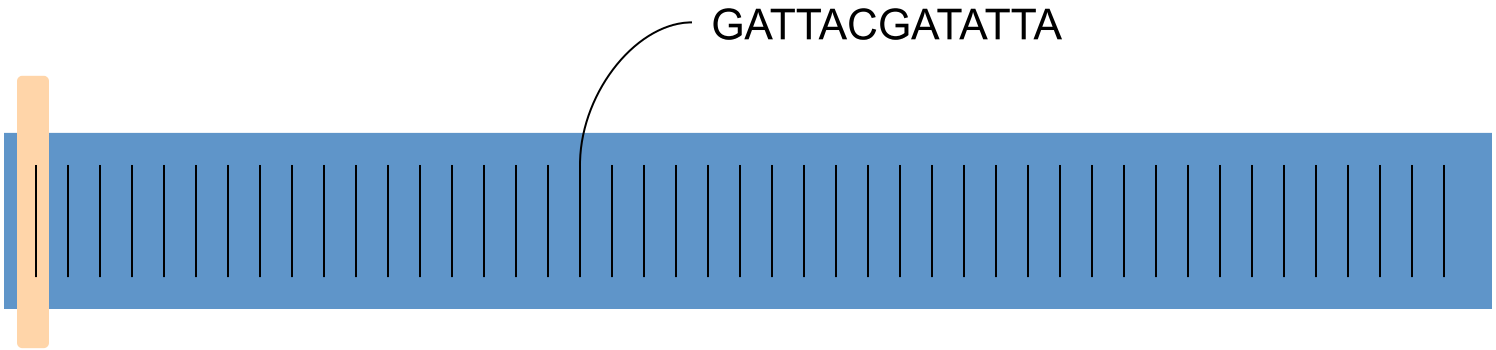
What Does Scalable Mean?

- Operationally:
  - In the past: **"Works even if data doesn't fit in main memory"**
  - Now: **"Can make use of 1000s of cheap computers"**
- Algorithmically:
  - In the past: **If you have N data items, you must do no more than *N^m* operations -- "polynomial time algorithms"**
  - Now: **If you have N data items, you must do no more than *N^m/k* operations, for some large *k***
    - Polynomial-time algorithms must be parallelized
  - Soon: **If you have N data items, you should do no more than *N * log(N)* operations**
    - As data sizes go up, you may only get one pass at the data
    - The data is streaming -- you better make that one pass count
    - Ex: Large Synoptic Survey Telescope (30TB / night)

Example: Find matching DNA sequences

- Given a set of sequences
- Find all sequences equal to "GATTACGATATTA"

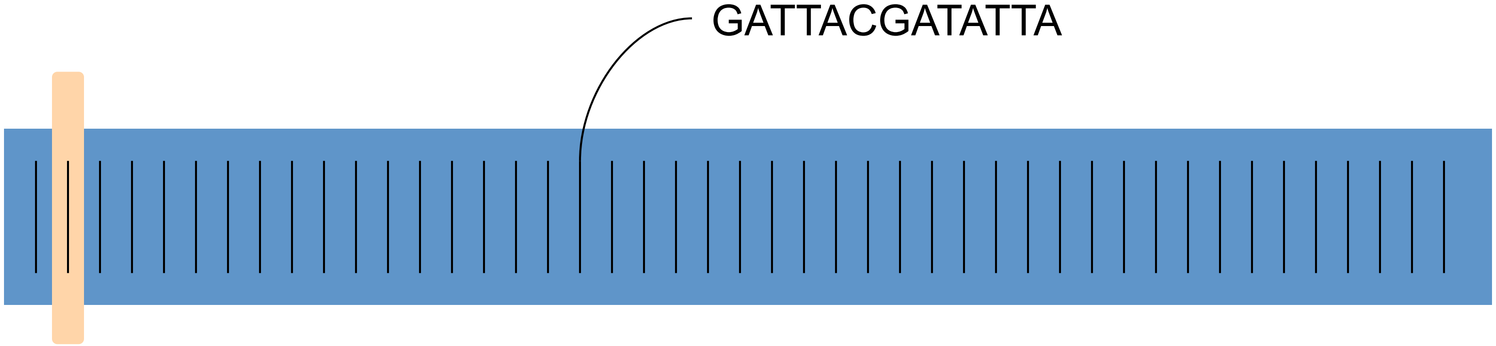

TACCTGCCGTAA
GATTACGATATTA



TACCTGCCGTAA = GATTACGATATTA?

No.

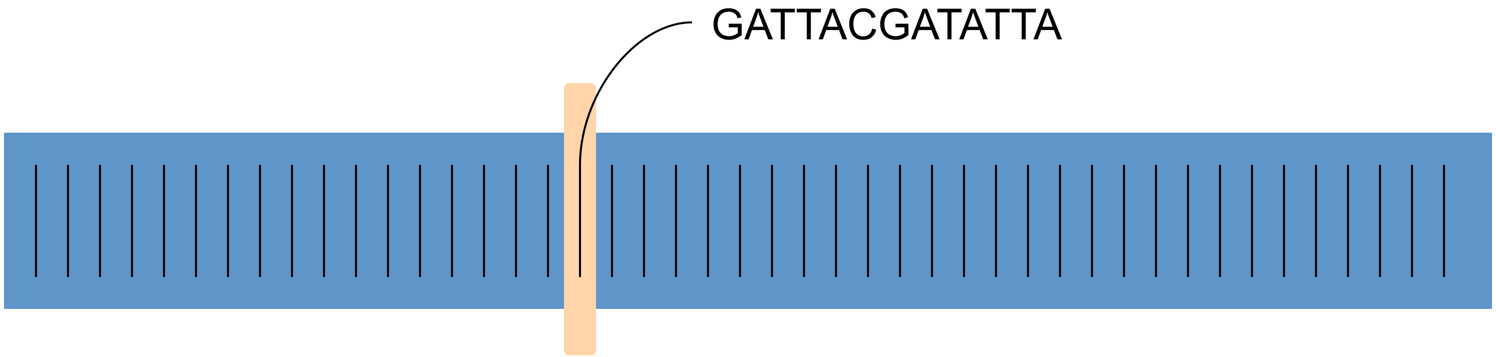
time = 0



CCCCCAATGAC = GATTACGATATTA?

No.

time = 1

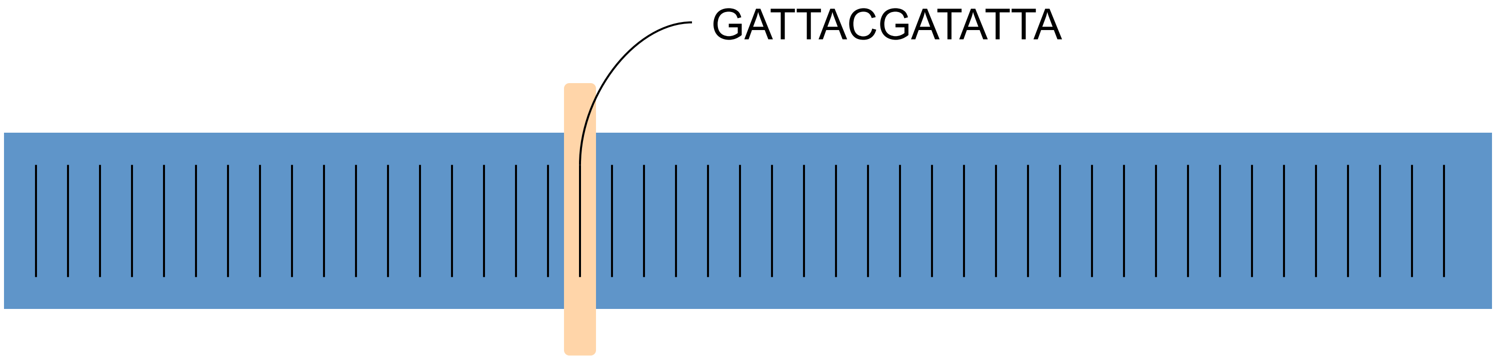


GATTACGATATTA contains GATTACGATATTA?

Yes!

Send it to the output.

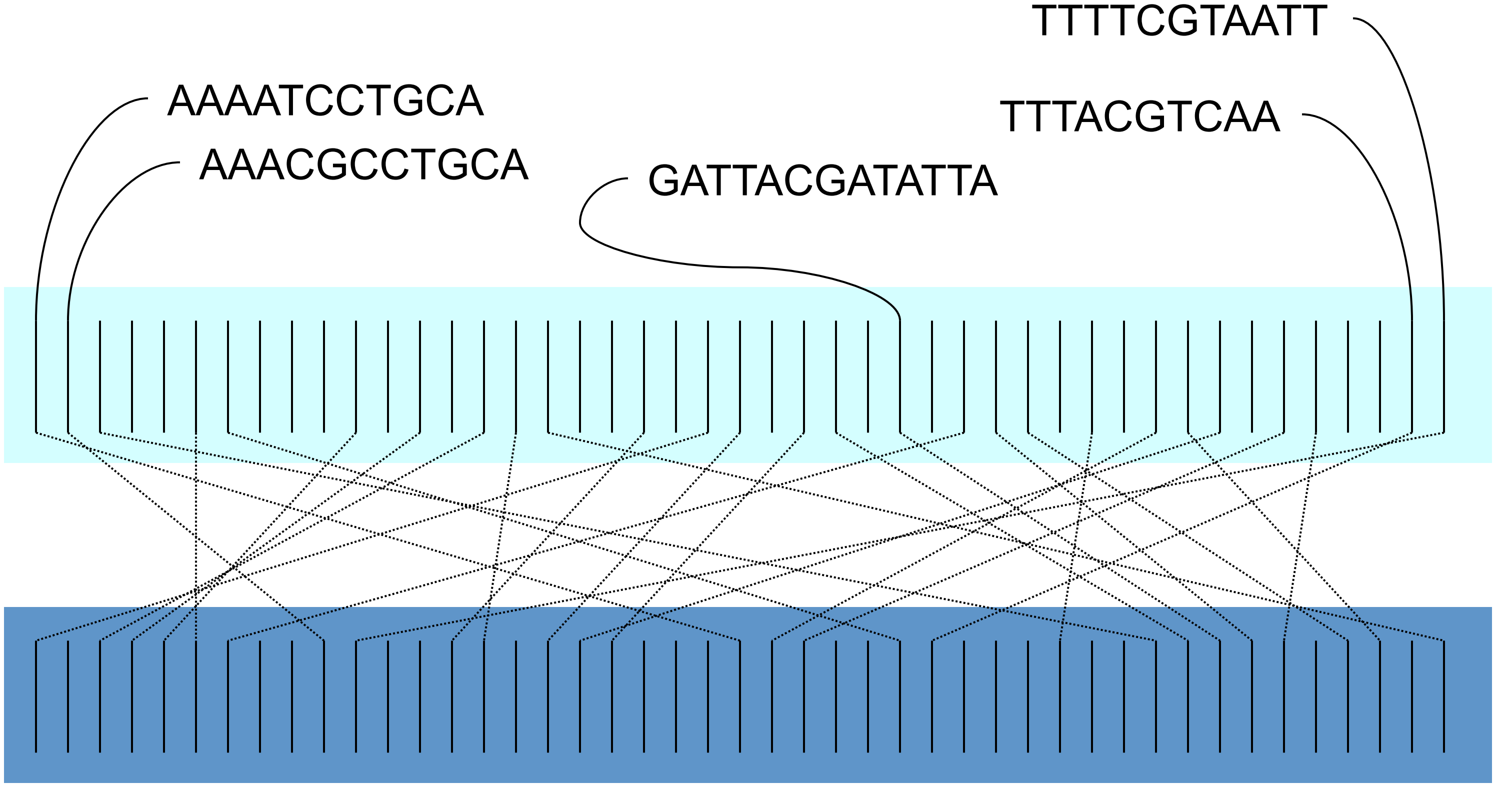
time = 17



40 records, 40 comparions

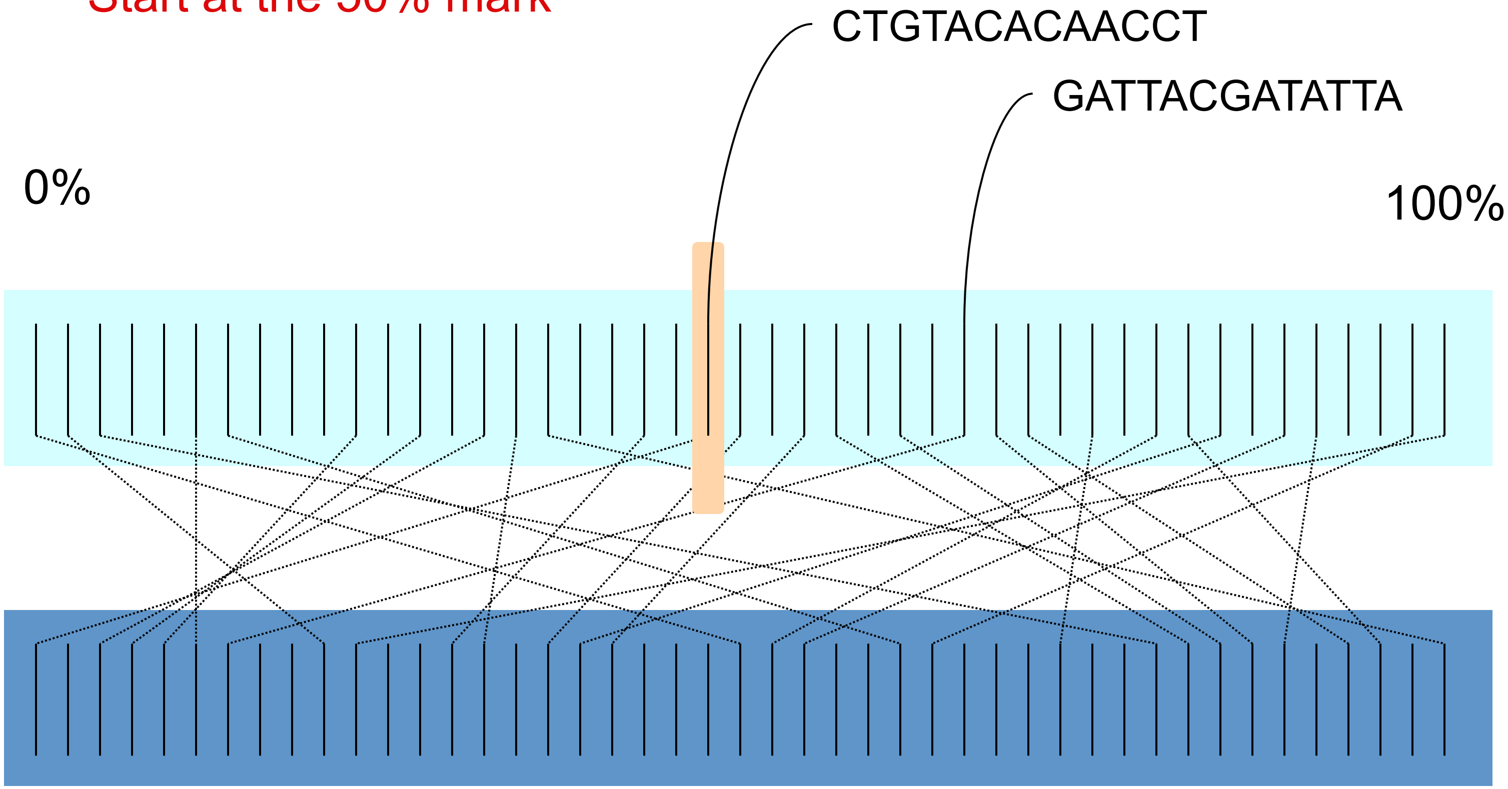
N records, *N* comparisons

The algorithmic complexity is order N: O(*N*)



What if we sort the sequences?

Start at the 50% mark

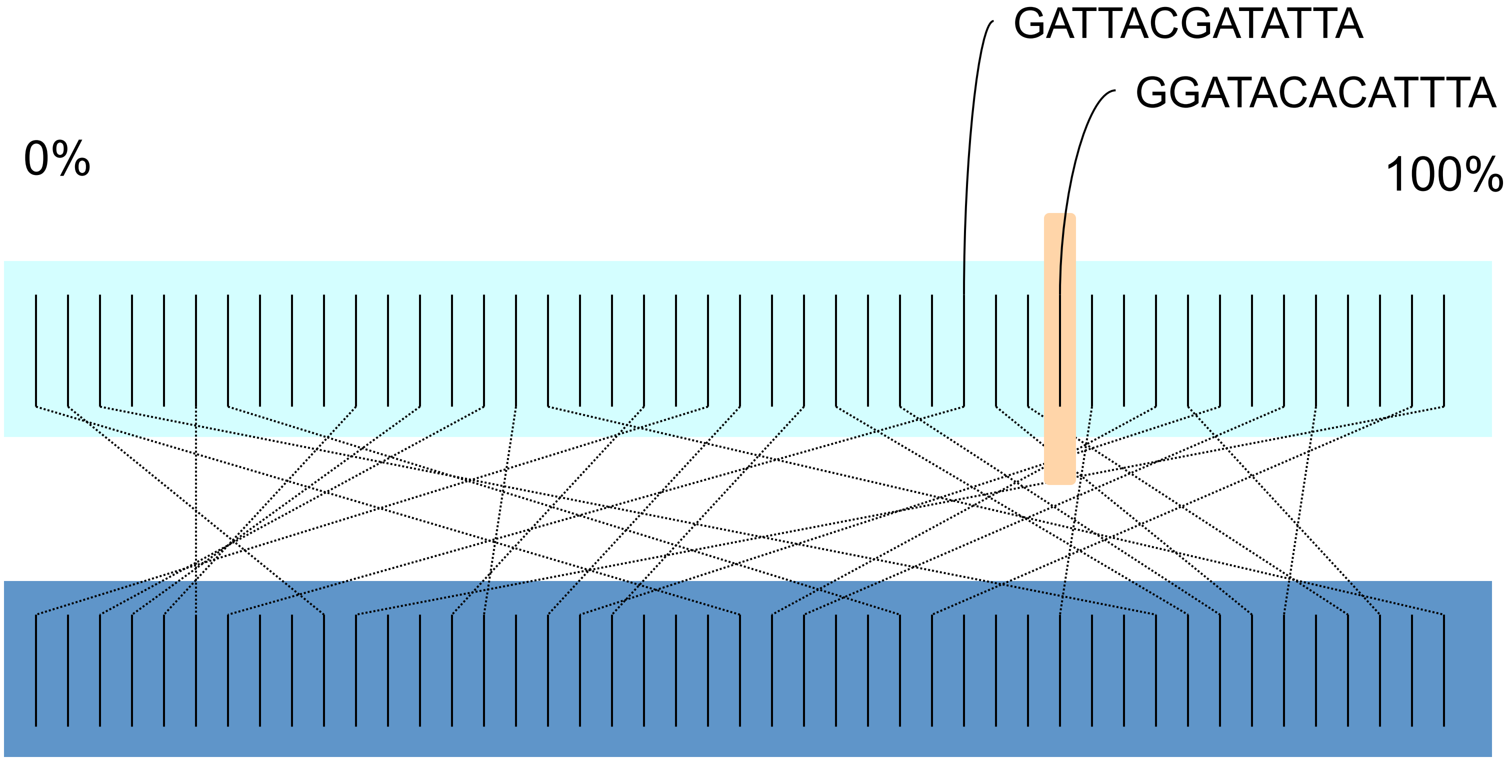


CTGTACACAACCT < GATTACGATATTA

time = 0

No match.

Skip to 75% mark

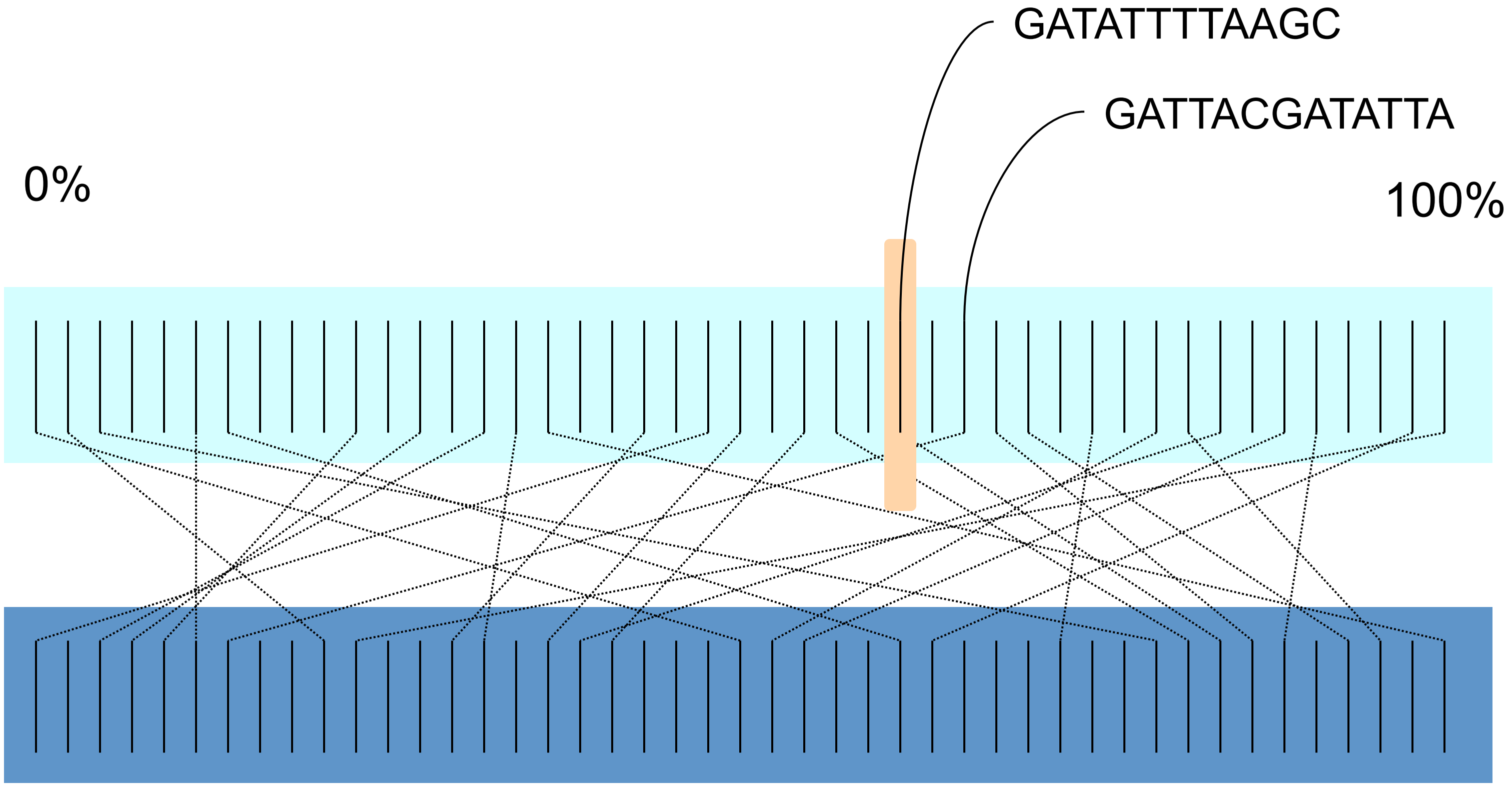


GGATACACATTTA > GATTACGATATTA

time = 1

No match.

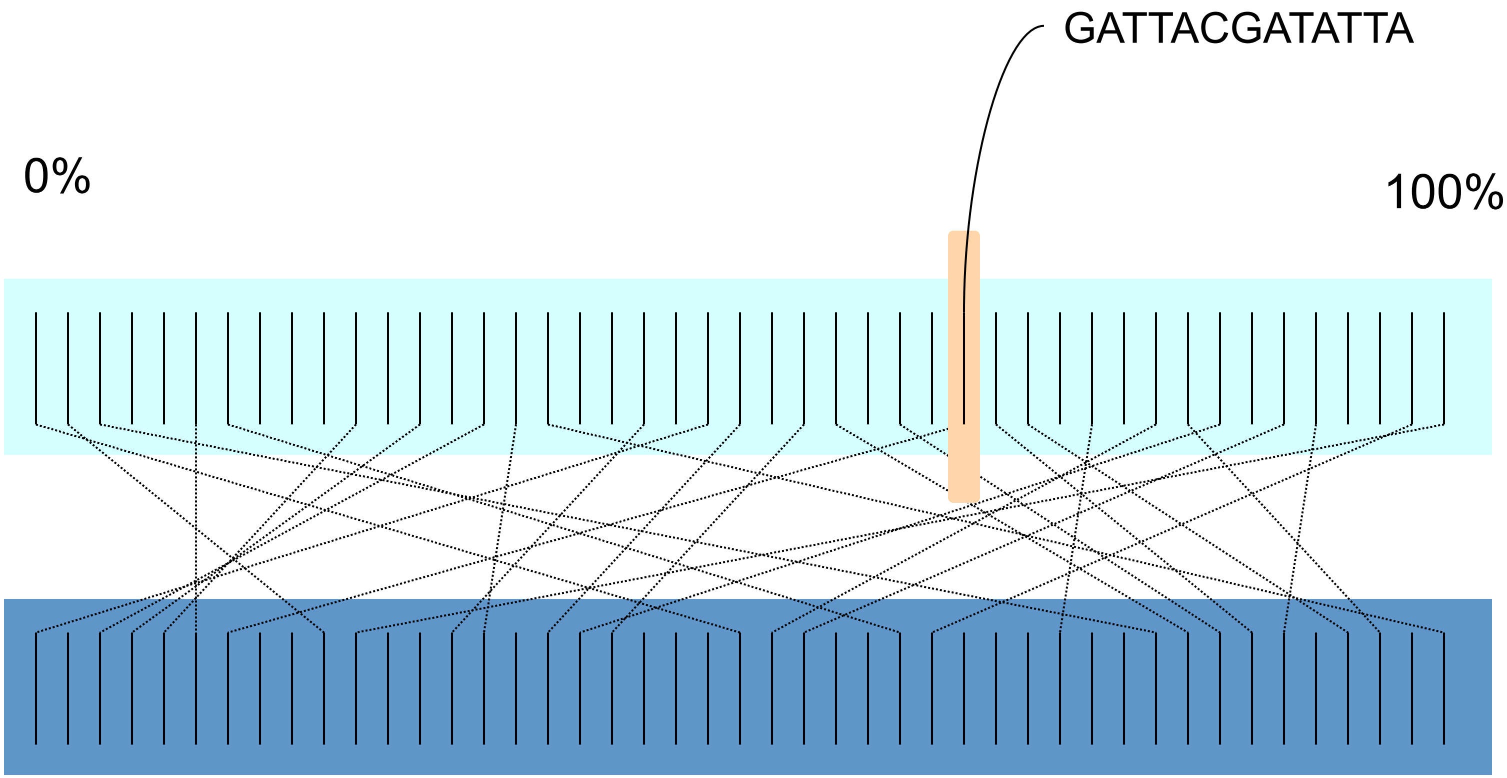
Go back to 62.5% mark



GATATTTTAAGC < GATTACGATATTA

No match.

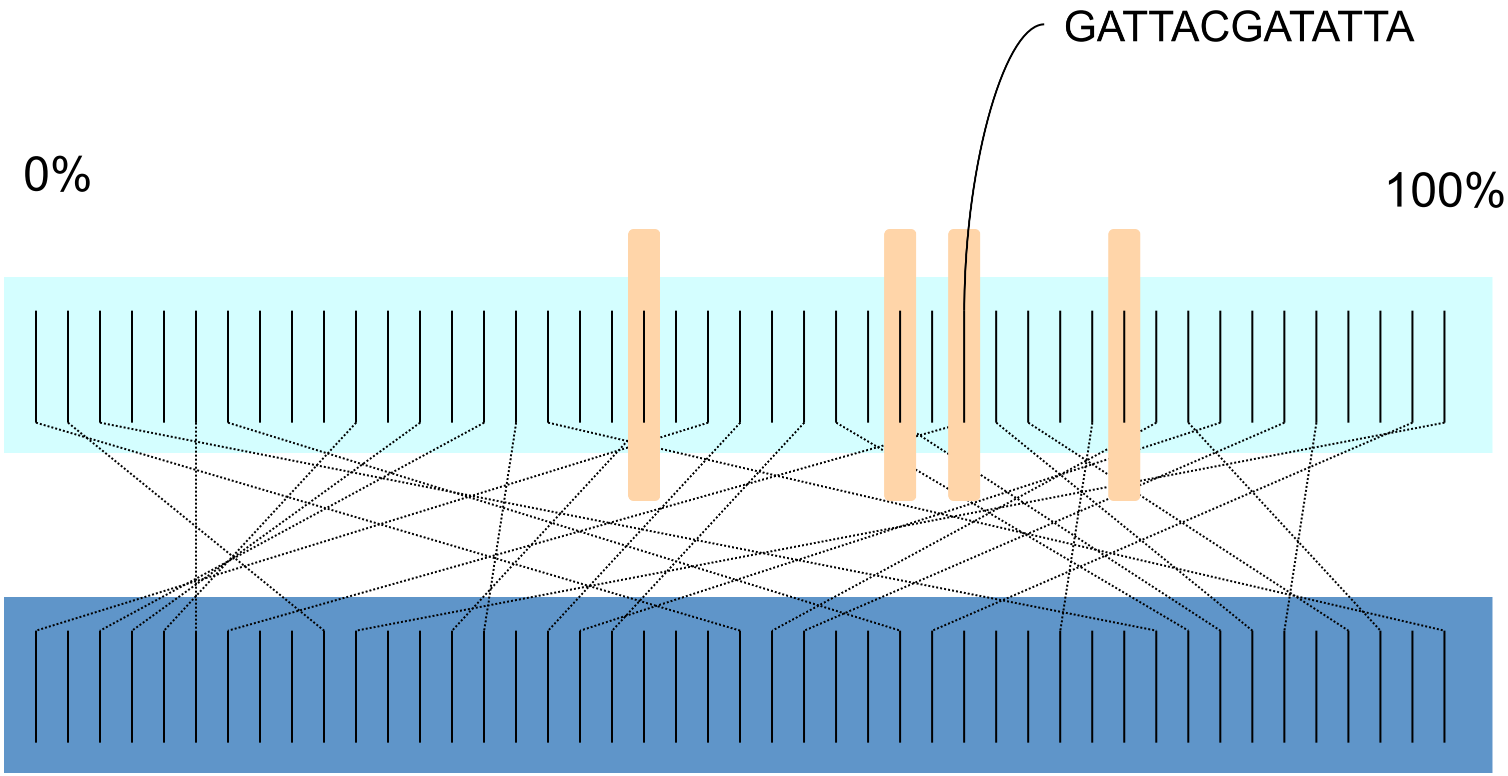
Skip back to 68.75% mark



GATTACGATATTA = GATTACGATATTA

Match!

Walk through the records until we fail to match.



How many comparisons did we do?

40 records, only 4 comparisons

N records, log(*N*) comparisons

This algorithm is O(log(*N*)) Far better scalability

Relational Databases

- Databases are good at "Needle in Haystack" problems:
  - **Extracting small results from big datasets**
  - **Transparently provide "old style" scalability**
  - **Your query will always* finish, regardless of dataset size.**

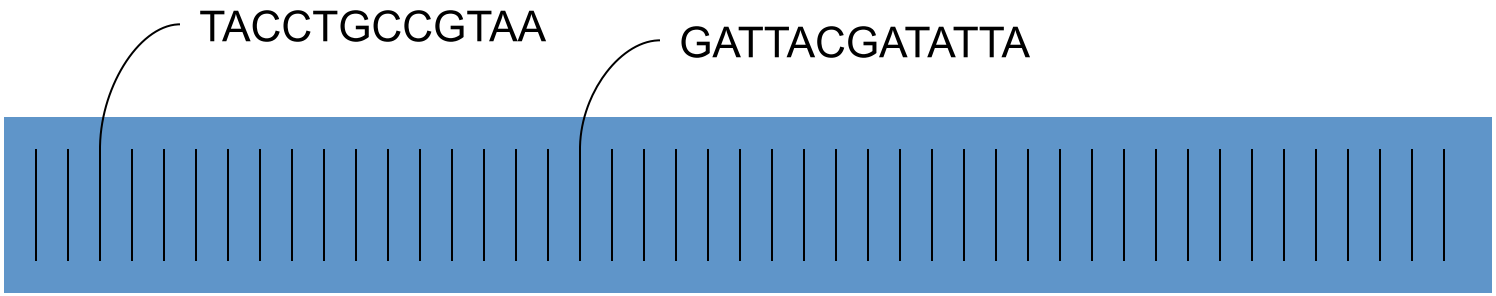
  - **Indexes are easily built and automatically used when appropriate**

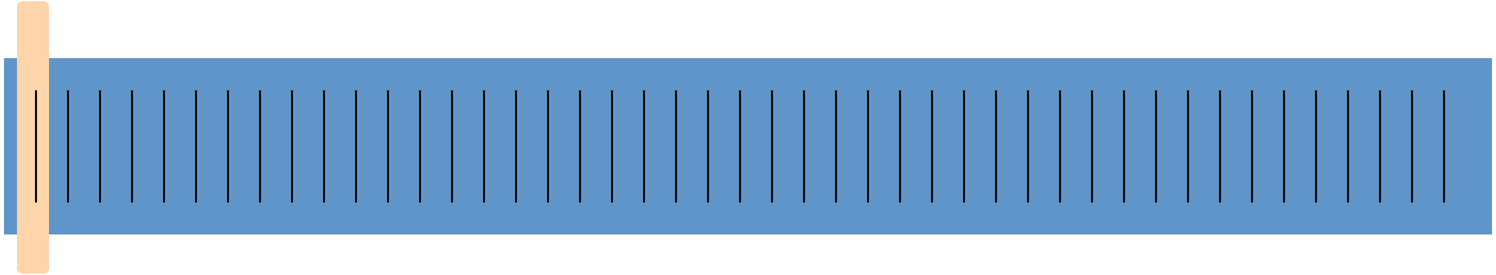
```
CREATE INDEX seq_idx ON sequence(seq);

SELECT seq
  FROM sequence
 WHERE seq = 'GATTACGATATTA';
```

***almost**

New task: Read Trimming

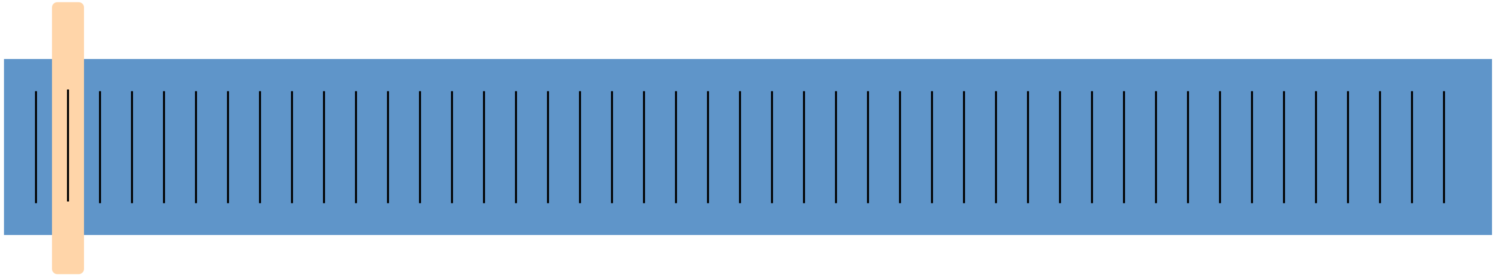
- Given a set of DNA sequences
- Trim the final *n* bps of each sequence
- Generate a new dataset


TACCTGCCGTAA
GATTACGATATTA



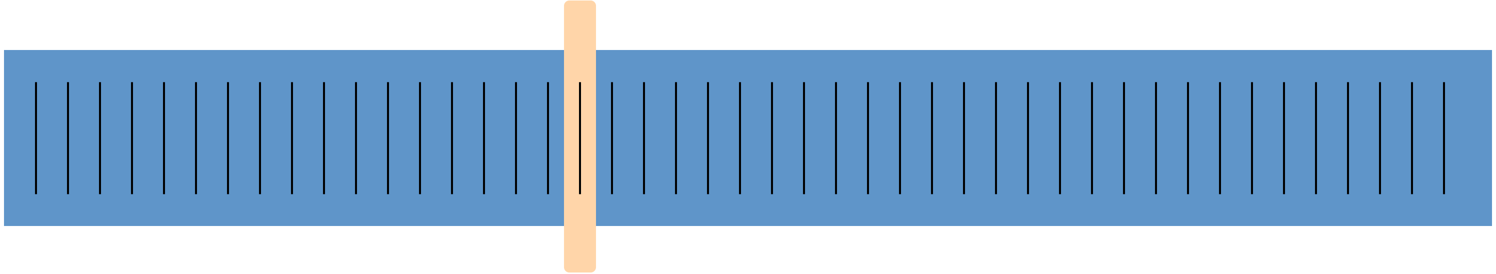
TACCTGCCGTAA becomes TACCT

time = 0



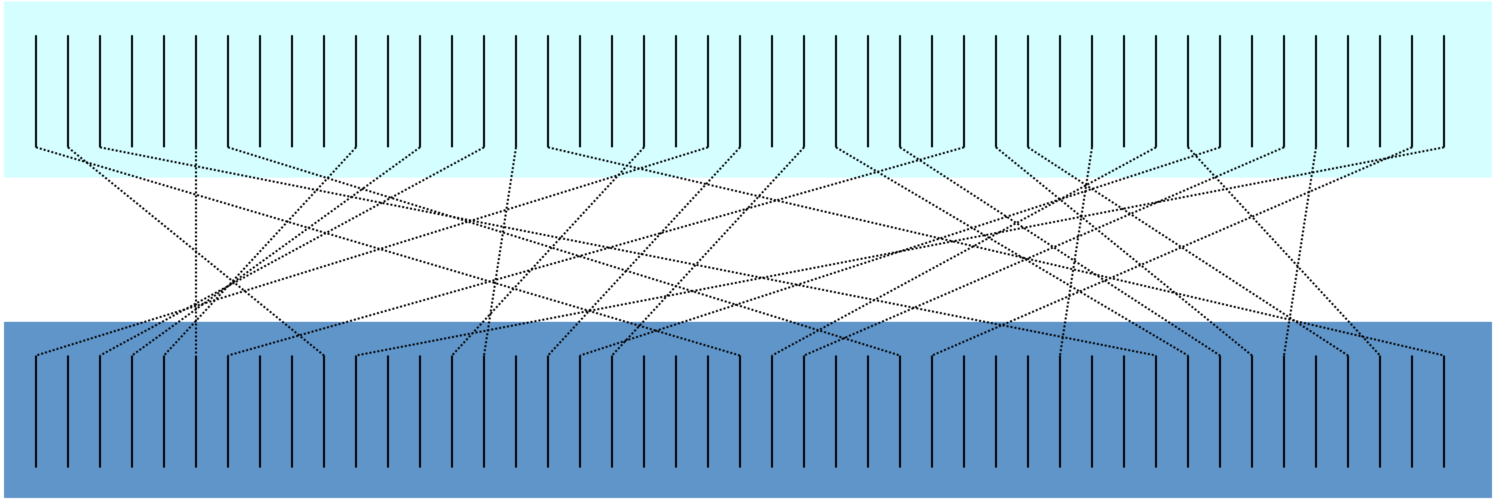
CCCCCAATGAC becomes CCCCC

time = 1



GATTACGATATTA becomes GATTA

time = 17

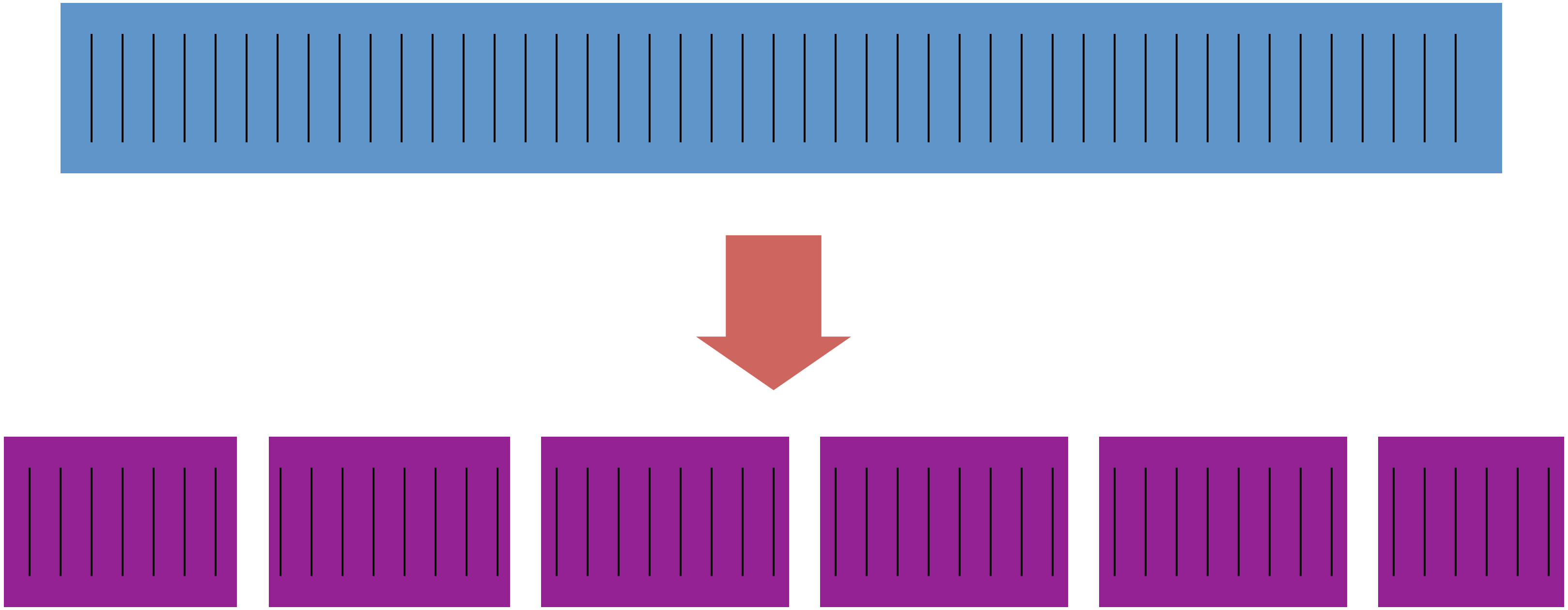


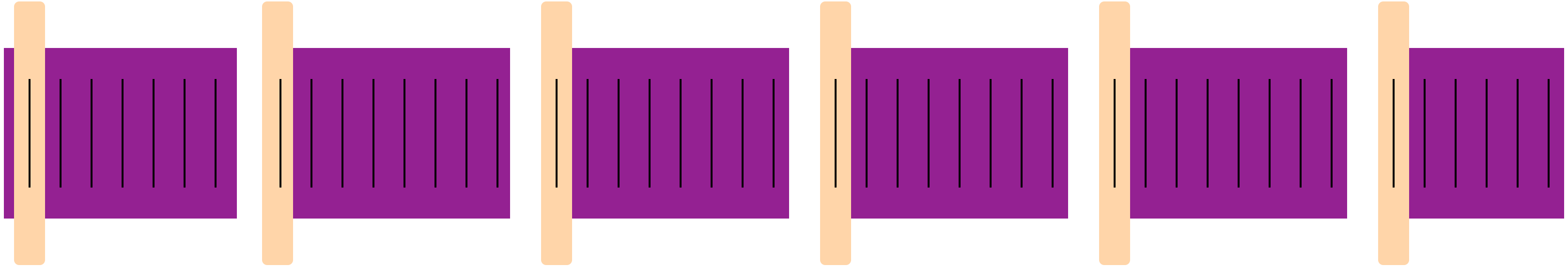
Can we use an index?

No. We have to touch every record no matter what.

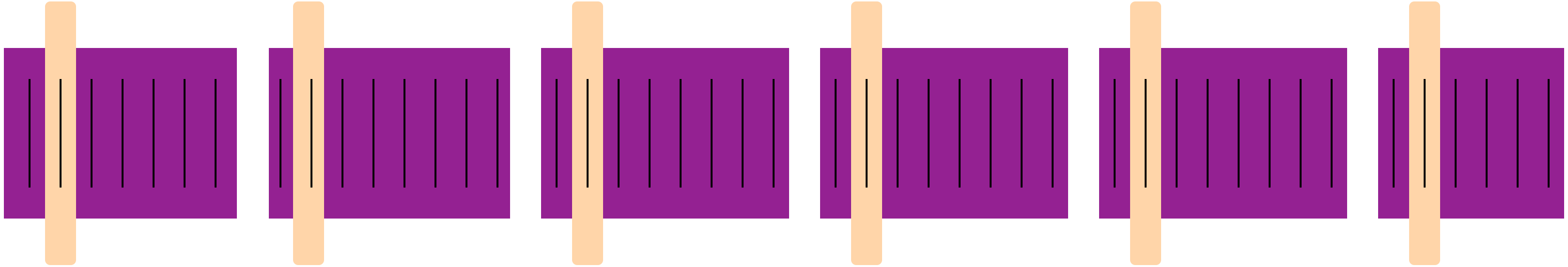
The task is fundamentally O(*N*)

Can we do any better?

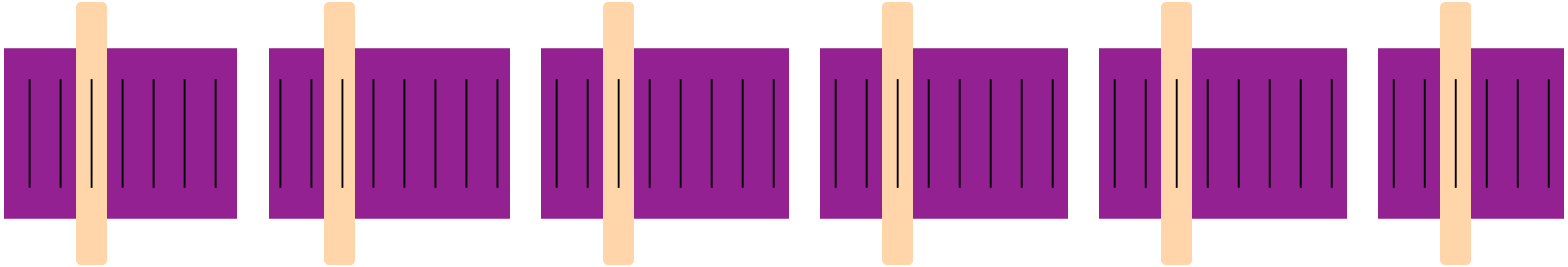




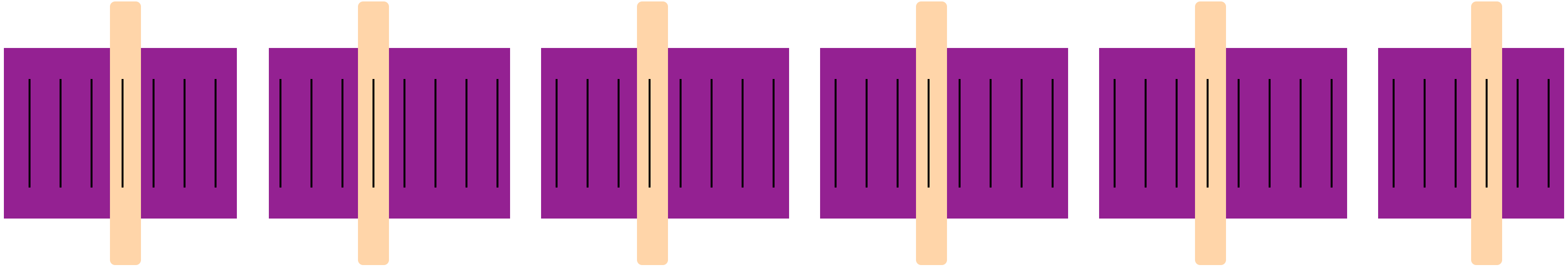
time = 0



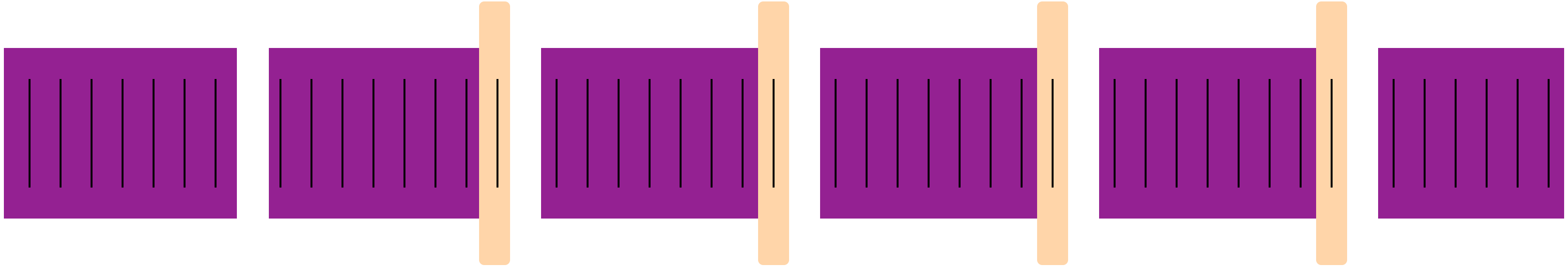
time = 1



time = 2



time = 3



How much time did this take?

7 cycles

40 records, 6 workers

time = 7

O(N/k)

Poll here:
https://pollev.com/billhowe