Ch 2 Fig. 2.14 Slides: tails for n = 0, works for n = 1

Parallelization: max value in set

<table>
<thead>
<tr>
<th>n x n grid</th>
</tr>
</thead>
</table>

Pattern-matching problem: DNA Ch. 2 slides

ATCG looking for TGC

```
ATTTT G CCC AT
```

"sliding window"

\[ T_1, T_2, T_3, T_4, T_5, T_6, T_7 \]

```
TGC
```

\[ m = 3 \]

\[ P_1, P_2, P_3 \]

Main idea: Scan the text characters left to right, for each possible starting location, try matching the pattern.
Parts of algorithm:

1. Slide pattern along the text, aligning it with each position.
2. Given a particular alignment, determine if there is a match at that location.

Get values for \( n \) and \( m \), size of text and pattern.
Set \( k \), the starting location for the attempted match, to 1.

While \( k \leq (n-m+1) \) do

\[
\begin{align*}
K &\leq (n-m+1) \quad \text{prevents pattern from "falling off" text} \\
P_1 &\quad P_2 \quad P_3 \\
T_k &\quad T_{k+1} \quad T_{k+2} \quad T_{k+3} \quad T_{k+m-1} \\
k+(m-1) &\leq n \\
k &\leq n-(m-1) \\
k &\leq n-m+1
\end{align*}
\]

Set value of \( i \) to 1.
Set value of MIS MATCH to NO.
While both \( i \leq m \) and (MIS MATCH = NO) do
If \( P_i \neq T_{k+(i-1)} \) then
Set MIS MATCH to YES.
Else
increment \( i \) by 1 (move to next character)
End loop.
If MISMATCH = NO then
  Print 'There is a match at position'
  Print value of k
  Increment k by 1
End loop
Stop

Finds all instances of pattern (does not stop if pattern found)

"nested loop"

If \( m = n \), algorithm will execute loop once

If \( m > n \), algorithm will not execute loops