

Exact matching

At what offsets does pattern P occur within text T ?



P : word



→ T : There would have been a time for such a word

Answer: offset 40

Exact matching

At what offsets does *pattern P* occur within *text T*?

```
>>> t = 'There would have been a time for such a word'  
>>> t.find('word')  
40
```


Exact matching

What's an algorithm for exact matching?


P: word

T: There would have been a time for such a word
word word word word word word word word word word
word word word word word word word word word word
word word word word word word word word word word
word word word word word word word word word word

Exact matching: naïve algorithm



```
def naive(p, t):
    occurrences = []
    for i in range(len(t) - len(p) + 1):  # Loop over alignments
        match = True
        for j in range(len(p)):          # Loop over characters
            if t[i+j] != p[j]:           # compare characters
                match = False            # mismatch; reject alignment
                break
        if match:
            occurrences.append(i)
    return occurrences
```



all chars matched; record

Exact matching: naïve algorithm

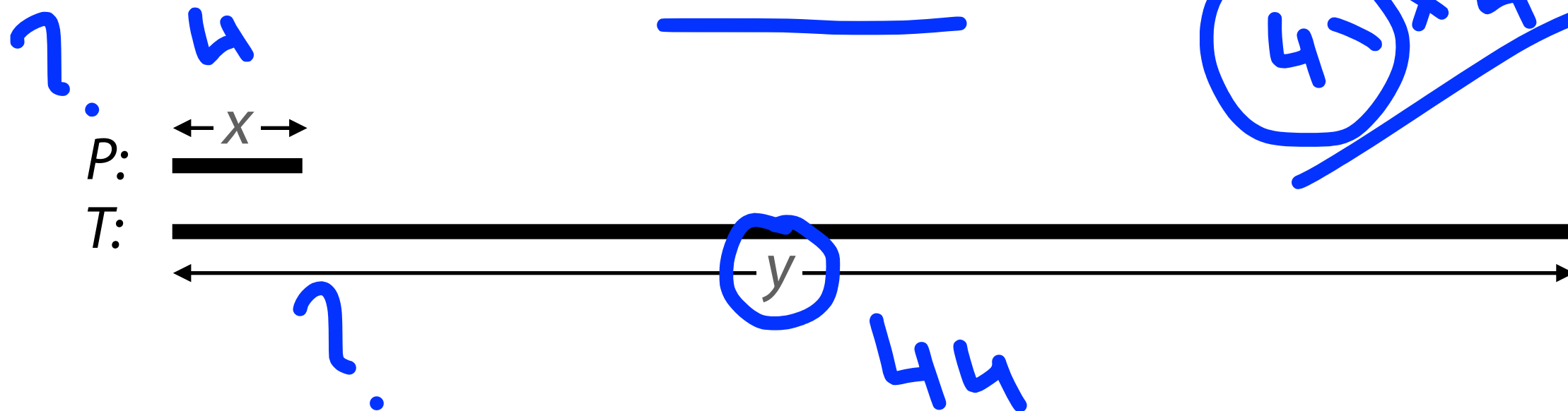
Let $x = |P|$, $y = |T|$

How many alignments are possible given x and y ?

$$O(y)$$

$$y - x + 1$$

$$4 \times 4$$



Exact matching: naïve algorithm

Let $x = |P|$, $y = |T|$

What's the greatest # character comparisons possible?

$$x(y - \underline{x} + 1)$$

$O(xy)$ ←

P : aaaa

T : aa

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa



Exact matching: naïve algorithm

Let $x = |P|$, $y = |T|$

What's the **least** # character comparisons possible?

$$y - x + 1$$

P : abbb

T : bbbbbb
abbb abbb abbb abbb abbb abbb abbb abbb abbb
abbb abbb abbb abbb abbb abbb abbb abbb
abbb abbb abbb abbb abbb abbb abbb abbb
abbb abbb abbb abbb abbb abbb abbb abbb
abbb abbb abbb abbb abbb abbb abbb abbb

Exact matching: naïve algorithm

How many character comparisons in this example?

P: word

T: There would have been a time for such a word
word word word word word word word word word
word word word word word word word word word
word word word word word word word word word
word word word word word word word word word
word word word word word word word word word

Exact matching: naïve algorithm

How many character comparisons in this example?

P: word

T: There would have been a time for such a word

word word word word word word word word word
word word word word word word word word
word word word word word word word word
word word word word word word word word
word word word word word word word word

40 mismatches + 6 matches = 46 character comparisons

Much closer to minimum (41) than maximum (164)