

CS 463: Homework 2

Kyle Guarco

May 12, 2023

1 Solution

Analyze the runtime of the following:

```
1  public static int search(String text, String pattern) {
2      int N = text.length();
3      int P = pattern.length();
4      for (int i = 0; i < N - P + 1; i++) {
5          int j;
6          for (j = 0; j < P; j++) {
7              if (text.charAt(i + j) != pattern.charAt(j)) {
8                  break;
9              }
10         }
11         if (j == P) {
12             return i;
13         }
14     }
15     return N;
16 }
```

Recall that a repeated sum of ones equals its bound: $\sum_{i=1}^n (1) = n$
A repeated sum of some constant C : $\sum_{i=1}^n (C) = Cn$

We write the loops in terms of summations (the 1 indicates a constant runtime):

$$\sum_{i=0}^{N-P} \left(\sum_{j=0}^{P-1} 1 \right) = \sum_{i=0}^{N-P} P$$

Reduce inner loop using ‘sum of ones’ formula

$$\sum_{i=0}^{N-P} P = P(N - P + 1)$$

Repeated sum of constant P

$$= -P^2 + NP + P$$

$\mathbf{O}(NP)$ ignoring the negative term.

Therefore the runtime of `search(String, String)` is $\mathbf{O}(nm)$.

2 Code

```
1 public class Main {
2
3     public static int search(String text, String pattern) {
4         int lengthOfText = text.length();
5         int lengthOfPattern = pattern.length();
6         // iterate through the whole text character by character
7         for (int i = 0; i < lengthOfText - lengthOfPattern+1; i++) {
8             int j;
9             // On every iteration we check whether the two characters are matching or not
10            for (j = 0; j < lengthOfPattern; j++) {
11                if (text.charAt(i + j) != pattern.charAt(j)) {
12                    break; // if mismatch: we break out
13                }
14            }
15            // We have found the pattern in the text because no mismatching character has been found
16            if (j == lengthOfPattern) {
17                // i is the first character in the match in this case it is the index of the character T
18                return i;
19            }
20        }
21        return lengthOfText; //The given text does not contain the given pattern
22    }
23
24    public static void main(String[] args) {
25        String text = "THIS IS A TEST";
26        String pattern = "TEST";
27
28        int result = search(text, pattern);
29
30        System.out.println("result: " + result);
31    }
32 }
```