Algorithms & Complexity 2/27/17

0145-344-001

Note Taker: Jai Punjwani

ANNOUNCEMENTS

Topic: Pattern Matching

PowerPoint: <http://home.adelphi.edu/~siegfried/cs344/344l7.pdf>

* Start by looking for first character of pattern in the query string, and looking for each subsequent character. If no match, start by looking for first character of pattern again.
* Very useful in computational biology (DNA sequence matching)
* Pattern matching can work for any type of pattern (i.e. bits)
* GREP used in Unix machines for pattern matching

**Analysis**

Worst-Case is NOT NECESSARILY when we don’t find any matches. In fact, if we find partial matches which eventually fail our pattern, we have to backtrack our index, which turns out to be worse.

ex:

String: aaaafj

pattern:aaaay

Initially we have an index at the first letter of our String. Our pattern will match the first four letters, but will fail the fifth. In that case, we will have to move our index to the **second** index. This is known as **backtracking** (perhaps not officially, but in terms of this algorithm). At the end of our pattern match search, we will have made **more than n** comparisons.

**Complexity:** O(m x n), where m = string length & n = pattern length

**Finite State Machines (FSM)**

* input String
* output Yes/No

