**CS251 Homework 5: Strings (II)**

**Out: April 13, 2018 @ 9:00 PM**

**Due: April 27, 2018 @ 11:59 PM**

**Submission Instructions: Please submit a typeset PDF on blackboard. For multiple choice questions, you must provide an explanation along with your answer. Answers without explanations will receive 0 points, even if correct.**

1. Finding patterns in DNA sequences is a common task in bioinformatics. A DNA sequence is composed of characters A, C, G and T representing adenine, cytosine, guanine and thymine, respectively. The KMP algorithm is used when the text and the pattern are not too long. Before running KMP, we must calculate the failure function of the pattern. We need to find the pattern “GACAGATGA” in a DNA sequence. Calculate the failure function for the given pattern. [2 points]

Define f() as failure function

f(G) = 0 (single character, 0)

f(GA) = 0 (No prefix)

f(GAC) = 0 (No prefix)

f(GACA) = 0 (No prefix)

f(GACAG) = 1 (G forms a prefix, increment to 1)

f(GACAGA) = 2 (GA forms a prefix, increment to 2)

f(GACAGAT) = 0 (No prefix)

f(GACAGATG) = 1 (G forms a prefix, increment to 1)

f(GACAGATGA) = 2 (GA forms a prefix, increment to 2)

so, it is:

000012012

1. Following the KMP algorithm, what is the number of comparisons required for finding the pattern “GACAGATGA” in “GGTACCCGACAGATGACAGA”? Help yourself with a drawing similar to the examples showed in class and paste it with your answer. [3 points]

String GACAGATGA

Prefix 000012012

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | G | G | T | A | C | C | C | G | A | C | A | G | A | T | G | A | C | A | G | A |
| 1 | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  | G | A | C | A | G | A | T | G | A |  |  |  |  |

1-2 mismatch at index1, prefix is 0, shift by 1, compared 2 characters

2-3 mismatch at index0, prefix is 0, shift by 1, compared 2 characters

3-4 mismatch at index0, prefix is 0, shift by 1, compared 1 character

4-5 mismatch at index0, prefix is 0, shift by 1, compared 1 character

5-6 mismatch at index0, prefix is 0, shift by 1, compared 1 character

6-7 mismatch at index0, prefix is 0, shift by 1, compared 1 character

7-8 mismatch at index0, prefix is 0, shift by 1, compared 1 character

Step8: all matches, compared 9 characters

So, there is 18 times that we compared characters

B

1. Consider a standard trie constructed from a chemistry book of 500 pages, each page having 2000 words (on average). Word lengths range from 1 letter to 1909 (There are long chemical names, and sometimes the author forgets to put hyphen to separate the groups. If you don’t believe me, check this: <https://en.wikipedia.org/wiki/Longest_word_in_English> ). We are interested in searching English words (so, our alphabet includes standard English alphabet, ignoring case, and numbers 0-9, totaling 36).  
   We want to search the word “methylhydroxide”. What will be the most number of comparisons to search the word? Show your work. [2 points]
   1. 68724
   2. 15
   3. 36
   4. 540

Because prefix is only stored once, and all strings or substrings who share same prefix, so the maximum comparisons are just its length which is 15.

D

1. How many tree nodes are there in a standard trie constructed from the following three words: albert, albany, albeit? Show your work by drawing the tree. [2 points]
   1. 7
   2. 9
   3. 11
   4. 12

A picture containing chain, metalware

Description generated with high confidence

C

1. A suffix trie is constructed from the word “sososo”. What will be the number of nodes in the trie (in uncompressed form)? Show your work by drawing the tree. [3 points]
   1. 10
   2. 14
   3. 18
   4. 22

A close up of a logo

Description generated with very high confidence

C

1. Consider the same suffix trie from the previous problem, but in compressed form. What will be the number of nodes in the compressed suffix trie? Show your work by drawing the tree. [3 points]
   1. 9
   2. 10
   3. 11
   4. 12

A close up of a mans face

Description generated with very high confidence

|  |
| --- |
| **Hint for 5 and 6:** For the example string “aka”, the suffix strings constructed will be:   1. aka$ 2. ka$ 3. a$   where $ is a special character used to denote the end of the string.  Then the uncompressed and compressed suffix trie will be the following:    [uncompressed suffix trie] [compressed suffix trie] |

1. Which uncompressed string is *the most inefficient* (in terms of the compression ratio) to be encoded using Run-Length Encoding? Which one is the *most efficient*? Show your work by providing Run-Length Encoding of each string. [2 points]
   1. AAAAAAAbbbXXXXXt
   2. AAAABBBAAACCC
   3. CGTACGTA
   4. CCGGTTAA

AAAAAAAbbbXXXXXt: 5A3bSX1t

AAAABBBAAACCC: 4A3B3A3C

CGTACGTA: two CGTA

CCGGTTAA: 2C2G2T2A

A is most efficient, can be shorted a lot in run-length encoding

D is most inefficient, still took 8 characters after encoding

B

1. How many connected components do we have after performing the following sequence of union operations on a set of 10 items? Show your work by drawing the tree. [2 points]

|  |
| --- |
| union(1,2), union(3,4), union(5,6), union(7,8), union(8,9), union(2,8), union(0,6), union(4,6) |

* 1. 1
  2. 2
  3. 3
  4. 4

A drawing of a person

Description generated with high confidence

1. Suppose we are working with a Union-Find data structure as described in the lecture slides. Suppose we consider a set of 10 items that are eventually joined into a single set via 9 union operations (assume that no path compression is applied). Can you have a sequence of union operations that will result in the root having three children? Explain why or why not.   
   [3 points]

No, we can’t make root having three children

In best case, after union operations, root will still have 4 children

A close up of text on a white background

Description generated with high confidence

1. Construct a regular expression and a deterministic finite automaton that accepts all strings using {a,b} where every accepted string ends with "ba" . Show your work. [2 points]

A close up of a logo

Description generated with very high confidence

1. Construct a deterministic finite automaton that accepts all non empty strings using {a,b} where every accepted string starts with "a" and the length of the string is a multiple of 4. Also provide its regular expression. Show your work. [3 points]

A screenshot of a cell phone

Description generated with very high confidence

1. a) Construct an Epsilon NFA for the regular expression a(aa|ab)\*ab. Show your work.   
    [2 points]

b) Construct a DFA for the NFA in (a). Show your work. [4 points]

*Total: 33 points.*