Connor Smith

Dr. Christina Zheng

BMI 550, F17

30 October 2017

HW 4

**Part 1:**

1. Suffix Array
2. Suffix Arrays are used for data compression and indexing, allowing for a more efficient and quick search of a large string for a specific substring. In Bioinformatics, this is useful for identifying the location of a sequence within a much larger sequence
3. Brute Force approach to searching for a string:
   1. Check eat letter of the whole string for the matching letter in the search string
   2. If a match is found, then check for the matching letter of the next string
   3. Continue until a match has been found for the entire search string
   4. Repeat until the entire string has been searched for occurrences
4. Suffix Array approach for searching for a string:
   1. Construct the suffix array, sorting in lexicographic order
   2. Check for a matching string at the beginning of each row
   3. Once found, continue to check until the first mismatch is found, at which point the search can stop
5. Advantages:
   1. Quick stopping: Once you find a match for your string, you can stop as soon as a row does not match due to the lexicographic ordering
   2. More efficient storage than a suffix tree: Has a lower space requirement than a suffix tree while maintaining the same functionality
6. Disadvantages:
   1. Storage: While the storage is more efficient that a suffix tree, the suffix array could still require much more storage than the string itself. In a human genome, this can be as high as 16 times the size of the original string

**Part 2:**

1. ACTGCTCGGCT
   1. Suffix Array:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | $ |  |  |  |  |  |  |  |  |  |  |  |
| **0** | A | C | T | G | C | T | C | G | G | C | T | $ |
| **6** | C | G | G | C | T | $ |  |  |  |  |  |  |
| **9** | C | T | $ |  |  |  |  |  |  |  |  |  |
| **4** | C | T | C | G | G | C | T | $ |  |  |  |  |
| **1** | C | T | G | C | T | C | G | G | C | T | $ |  |
| **8** | G | C | T | $ |  |  |  |  |  |  |  |  |
| **3** | G | C | T | C | G | G | C | T | $ |  |  |  |
| **7** | G | G | C | T | $ |  |  |  |  |  |  |  |
| **10** | T | $ |  |  |  |  |  |  |  |  |  |  |
| **2** | T | G | C | T | C | G | G | C | T | $ |  |  |
| **5** | T | C | G | G | C | T | $ |  |  |  |  |  |

* 1. BWT:

Unsorted:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | C | T | G | C | T | C | G | G | C | T | $ |
| $ | A | C | T | G | C | T | C | G | G | C | T |
| T | $ | A | C | T | G | C | T | C | G | G | C |
| C | T | $ | A | C | T | G | C | T | C | G | G |
| G | C | T | $ | A | C | T | G | C | T | C | G |
| G | G | C | T | $ | A | C | T | G | C | T | C |
| C | G | G | C | T | $ | A | C | T | G | C | T |
| T | C | G | G | C | T | $ | A | C | T | G | C |
| C | T | C | G | G | C | T | $ | A | C | T | G |
| G | C | T | C | G | G | C | T | $ | A | C | T |
| T | G | C | T | C | G | G | C | T | $ | A | C |
| C | T | G | C | T | C | G | G | C | T | $ | A |

Sorted:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $ | A | C | T | G | C | T | C | G | G | C | **T** |
| A | C | T | G | C | T | C | G | G | C | T | **$** |
| C | G | G | C | T | $ | A | C | T | G | C | **T** |
| C | T | $ | A | C | T | G | C | T | C | G | **G** |
| C | T | C | G | G | C | T | $ | A | C | T | **G** |
| C | T | G | C | T | C | G | G | C | T | $ | **A** |
| G | C | T | $ | A | C | T | G | C | T | C | **G** |
| G | C | T | C | G | G | C | T | $ | A | C | **T** |
| G | G | C | T | $ | A | C | T | G | C | T | **C** |
| T | $ | A | C | T | G | C | T | C | G | G | **C** |
| T | C | G | G | C | T | $ | A | C | T | G | **C** |
| T | G | C | T | C | G | G | C | T | $ | A | **C** |

Result:

**T$TGGAGTCCCC**

1. It does not, as node 7 creates an island with nodes 9 and 10 that would require you to return to node 7 after having already visited it. Trying to do a Hamiltonian cycle on this graph would result in visiting node 7 twice (if you start with a different node) or three times (if you start with 7).