NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CX2001 Algorithms **2020/21 Semester 1**

Project 1: Searching Algorithm

School of Computer Science and Engineering *Nanyang Technological University*

1.1 Problem

DNA and protein sequence searching is one of fundamental problems in bioinformatics research. A nucleic acid sequence is a series of a set of five letters that indicate the order of nucleotides forming alleles within a DNA (G, A, C, T) or RNA (G, A, C, U) molecule. The range of sequences in size is from a few nucleotides to billions of base pairs. Searching for a query sequence in a nucleic acid sequence is a time-consuming process.

In this project, we first would like to study the time complexity of a brute force sequential search for exact occurrences of a query sequence in a nucleic acid sequence. As an example, we would like to search for a query sequence (TTTATACCTTCC) in the corovirus genome sequence below:

>NC_ 045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGATCTGTT
CTCTAAACGAACTTTAAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCACGCA
GTATAATTAATAACTAATTACTGTCGTTGACAGGACACGAGTAACTCGTCTATCTTCTGCAGG
CTGCTTACGGTTTCGTCCGTGTTGCAGCCGATCATCAGCACATCTAGGTTTCGTCCGGGTGTG
ACCGAAAGGTAAGATGGAGAGCCTTGTCCCTGGTTTCAACGAGAAAACACACGTCCAACTCAG
TTTGCCTGTTTTACAGGTTCGCGACGTGCTCGTACGTGGCTTTGGAGACTCCGTGGAGGAGGT
CTTATCAGAGGCACGTCAACATCTTAAAGATGGCACTTGTGGCTTAGTAGAAGTTGAAAAAGG
CGTTTTGCCTCAACTTGAACAGCCCTATGTGTTCATCAAACGTTCGGATGCTCGAACTGCACC
TCATGGTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAGGCATTCAGTACGGTCGTAGTGG
TGAGACACTTGGTGTCCTTGTCCCTCATGTGGGCGAAATACCAGTGGCTTACCGCAAGGTTCT
TCTTCGTAAGAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATT
TGACTTAGGCGACGAGCTTGGCACTGATCCTTATGAAGATTTTCAAGAAAACTGGAACACTAA
ACATAGCAGTGGTGTTACCCGTGAACTCATGCGTGAGCTTAACGGAGGGGCATACACTCGCTA
....

The algorithm should return all the positions (indexes) of exact occurrences of the query in the genome sequence. In this example, there is only an occurrence at the 9^{th} position. If there is no any exact match, the algorithm should also return an indicator like "empty" or "no occurrence" etc.

It is not surprising that the brute force searching is very slow especially searching in a long genome sequence. Next, each group is free to propose up to two other algorithms for the searching process. The time complexity of the proposed algorithms has to be better than the brute-force searching's. In the evaluation, lab supervisor will use the brute-force searching as a baseline.

In this project, students are not only required to implement algorithms, but also analyse the time complexity of their algorithms. The design and analysis of the algorithms have to be correct and complete. The implementation program has to be the same algorithm proposed in the report.

1.2 Data Resource

Genome sequences can be downloaded from the Assembly in Genomic FASTA format (.fna). Details of the download instruction can be found in https://www.ncbi.nlm.nih.gov/genome/doc/ftpfaq/. Students are just required to download a few sequences for experiments and demonstrations. Lab supervisor reserves the right to test students' work by other genome sequences in the genomic fasta format (.fna). It is noted that the file size can range from a few kilobytes (virus genome) to a few gigabytes (homo sapiens genome). Some groups may not be able to handle large file size. The limitation of the file size should be indicated in your report.

1.3 Implementation

The implementation codes of a brute-force sequential search is the minimum requirement in this project. The input arguments are a query sequence and a genome sequence. The genome sequence is stored in a .fna file. The interface should be user-friendly. So that, lab supervisor can easily change other sequences to test the implementation program and code recompilation is not required for changing the inputs. Outputs of the program are all positions of exact occurrences in the genome sequence. Students are allowed to write any program for conducting some experiments for their algorithm design. These work has to be well documented and submitted to their lab supervisors for verification if any experimental results is reported in the 5-page report.

1.4 Design and Analysis of Algorithm

Each group is required to analyse all the implemented algorithms in the project. The assessment will be based on students' analytical skills, correctness of their analysis and empirical analysis work. No comparison with the state-of-the-art work will be considered in the assessment. Students should not worry that the time complexity of their proposed algorithms does not outperform any published research work. However, reasonable comparison among groups' work is possible during the assessment. The lab supervisor will evaluate your work based on the correctness and completeness of analysis of algorithm and originality of the proposed algorithms. All reference materials have to stated clearly in the report. Students is reminded that plagiarism is a very serious academic offence. If any student may not knowingly intend to plagiarise, but that should not be used as an excuse for plagiarism. Students should seek clarification from lecturers or lab supervisors if they are unsure whether or not they are plagiarising the work of another person.