Data Structure and Algorithm

Homework #6

Due: 2:00pm, Thursday, June 14, 2012

TA email: dsa1@csie.ntu.edu.tw

=== Homework submission instructions ===

- For Problem 1, submit your source code, a shell script to compile the source, and a brief documentation to the SVN server (katrina.csie.ntu.edu.tw). You should create a new folder "hw6" and put these three files in it.
- The filenames of the source code, the shell script, and the documentation file should be "genome.c", "compile.sh", and "report.txt", respectively. **The shell script should compile your source codes to generate an executable binary with the filename 'genome'**. You will get some penalties in your grade if your submission do not follow the naming rule.
- The documentation file should be in plain text format (.txt file). In the documentation file you should explain how your code works, and anything you would like to convey to the TAs.
- For Problem 2 to Problem 5, submit the answers through the CEIBA system (electronic copy) or to the TA in R508 (hard copy).
- Each student may only choose to submit the homework in one way; either all as hard copies or all through CEIBA except the programming assignment. If you submit your homework partially in one way and partially in the other way, you might only get the score of the part submitted as hard copies or the part submitted through CEIBA (the part that the TA chooses).
- If you choose to submit the answers of the writing problems through CEIBA, please combine the answers of all writing problems into only ONE file in the doc/docx or pdf format; otherwise, you might only get the score of one of the files (the one that the TA chooses). In addition, **do NOT forget to specify your name and student ID** in your submitted file.
- Discussions with others are encouraged. However, you should write down your solutions by your own words. In addition, you have to cite the sources you consulted from or people you discussed with on the first page of your solution to that problem. The TA can deduct up to 100% of the score for any problem without the citation.
- No late submission of the homework will be given any score (for that portion).

Problem 1. (40%) Begun formally in 1990, the U.S. Human Genome Project was a 13-year effort. A key component of the Human Genome Project was that the entire human genome sequence would

be made freely accessible to the public. Nowadays, approximately 20,000-25,000 genes sequences in human DNA are identified.

"Exons" are important parts in genes that codes for a specific portion of the complete protein and those proteins have be found to cause some diseases. For example, *BRCA2* located on chromosome 13 is a well-known gene related with breast cancer. Scientists aim at finding disease-related genes. To be more specific, the goal is to find the position of exons in every gene. In this problem, you will write a program to find the position of some exons of a gene. You are required to use the Knuth-Morris-Pratt string matching algorithm to accomplish this task.

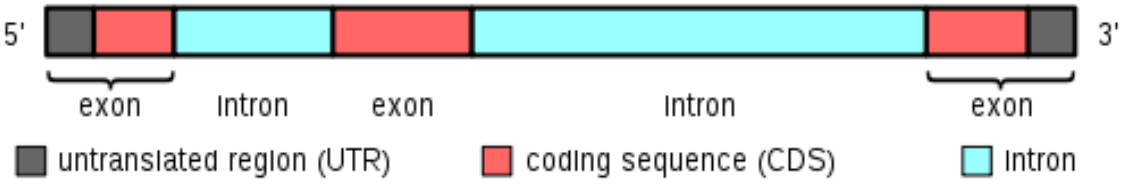


Figure 1: Figure 1. An illustration of gene structure.

Input:

The first line contains a chromosome sequence. The length of the sequence is within 250,000,000 characters. The next line is a number n ($1 \leq n \leq 10$), indicating that there are n exons to be matched. The following n lines are the exons, each with length no longer than 5000 characters.

Output:

For each exon, you have to output 2 lines.

Failure function (in KMP algorithm)

Matched index

The failure function is represented by a sequence of numbers, separating by a space. The index of every sequence starts from 0. The matched index is the index of the FIRST occurrence of the exon in the chromosome sequence. Output -1 for the case in which no match is found. You can reference to the sample input and output.

Sample Input:

```
AGAAATGACTTCGAATTCCTCCAGGAGGC
3
AATTC
GAG
AGTAGCAGTC
```

Sample Output:

```
-1 0 -1 -1 -1
13
-1 -1 0
24
-1 -1 -1 0 1 -1 0 1 2 -1
-1
```

Additional sample test data will be made available on the course web site.

Please write a program to solve this problem. Please also submit a report in which you give clear description of your code.

Problem 2. Sorting (16%)

1. (8%) Design an algorithm to sort an array of integers, where different integers have different numbers of digits. The algorithm should run in $O(n)$ time, where n is the total number of digits over all the numbers in the array.

2. (8%) Design an algorithm to sort an array of strings in standard alphabetical order (e.g. $a < ab < b$), where strings in the array may have different numbers of characters. The algorithm should run in $O(n)$ time, where n is the total number of characters over all the strings in the array.

Problem 3. Hashing (20%)

1. (10%) Consider a version of the division method in which $h(k) = k \bmod m$, where $m = 2^p - 1$ and k is a character string interpreted in radix 2^p. Assume that the character string $x = \{x_0, x_1, ..., x_n\}$ where x_i is one of the characters in x, $0 \leq i \leq n$, then

$$h(x) = \left(\sum_{i=0}^{n-1} f(x_i) * (2^{ip}) \right) \mod (2^p - 1) \tag{1}$$

   where f is a function to map a character to a number, e.g., its ASCII code. Show that if we can derive string x from string y by permuting its characters, then x and y hash to the same value.

2. (10%) Insert the keys $\{17, 33, 8, 36, 39, 31, 88\}$ in the given order into a hash table of length $m = 11$ using open addressing with double hashing. Assume that the primary hash function is $h_1(k) = k \bmod m$ and the secondary hash function is $h_2(k) = 1 + (k \bmod (m - 1))$. Illustrate how these keys are inserted into the hash table in a step-by-step manner after every insertion.

Problem 4. Quadratic Probing (20%)

Suppose that we are given a key k to search for in an initially empty hash table with positions $0, \ldots, m-1$, and suppose that we have a hash function h mapping the key space into the set $\{0, 1, \ldots, m-1\}$.

The searching scheme is as follows:

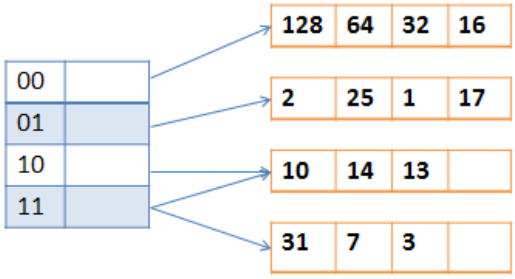
1. Compute the value $j = h(k)$, and set $i = 0$.
2. Probe in position j for the desired key k. If you find it, or if this position is empty, terminate the search.
3. Set $i = i + 1$. If i now equals m, the table is full, so terminate the search. Otherwise, set $j = (i + j) \mod m$, and return to the second step.

Assume that m is a power of 2. Please answer the following questions:

1. (10%) Show that this scheme is an instance of the general "quadratic probing" scheme by exhibiting the appropriate constants c_1 and c_2 for equation $h(k, i) = (h'(k) + c_1 i + c_2 i^2) \mod m$ (Cormen p.272).
2. (10%) Prove that this algorithm examines every table position in the worst case.

Problem 5. Dynamic Hashing (10%)

1. (4%) The hash table below uses the dynamic hashing with the directory, and each table entry can hold up to 4 keys. The hash function $h(k, d)$ is defined as the last d bits of binary representation of key k. Is there anything wrong with the table? Please fix the problem(s).



2. (2%) After correcting the table, show the content of the table after inserting 22 into it.
3. (2%) Continuing from the last question, show the content of the table after inserting 33 into it.
4. (2%) Continuing from the last question, show the content of the table after inserting 12 into it.