MATH-5170 Group Project Report

By: Qianhui Guo, Ziyu Jin, Wei Li, Ying Xue

December 14th, 2018

Book Identification System

In this project, we use python to create a book identification system, where each book has its own unique 8-character identifier and over 28 million books, 7 more item collections and nearly 1000 years are allowed to be expanded into the library catalogue in the future.

During the process, there are mainly 3 steps to achieve creating unique keys for each book:

1. Data Storage

In the first part, we load the library data into the python program and store all the information of each book for further use, including creating keys with the information of publication year and item collections, as well as accessing the book information in book inquiry system by using the keys we created.

2. Book Identifier Encoding

Next, we apply the information of publication year and item collections to creating the book identification system with the method of conversion of number systems.

2.1 Character Set

To ensure that the characters we use are characteristically distinct, we select totally 31 characters in a set from all the numbers and letters, removing those characters which are not characteristically distinct (‘0’,’O’,’I’,’V’,’U’). Here shows the 31-character charset: {'1', '2',

'3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'W', 'X', 'Y', 'Z'}

2.2 Information Division and Encoding

Besides, as each identifier should contain 8 characters with the information of publication year and item collections, we divide the 8 characters into 3 parts representing 3 different information (counting number, publication year and item collections). Then we aim to find out a number to represent the information in each part and thus, the combination of these 3 numbers can only represent a unique book. In the following, the process of finding a number for each part is shown. In the first part, each book is given a different counting number when counting from 1 to the end (250966) in the data. In the 2nd part, we first set the starting publication years from 1600, then we calculate the difference between the starting years and the exact publication year of the book. In the last part, we create a dictionary to store the item collections and assign a number for the item collection according to the order of is the first appearance in the dictionary.

After gaining the three numbers, what we need to do is to condense the number into a shorter code. So we apply the method of conversion of number systems respectively in each part, converting the number from the decimal system to base-31 system. Instead of using only 10 numbers to represent a number, we use 31 characters in the charset to represent a number to shorten the length of the number.

2.3 Information Capacity(Expansion)

As what's required for the identifier, we have to use exactly 8 characters to represent 1 million distinct books with their publication years and item collections. Because the range of variation of books is required up to 1 million, which is much larger than the other two parts

(publication years and item collections). We start to figure out the number of digits (characters) needed to encode the counting number first. As we discussed above, we have 31 distinct characters to choose for each digit, so the number of the digits we need is log31 (1,000,000), equal to 4.023, which means that at least 5 characters should be assigned to encode the counting number from 1 to 1 million. In this case, we designated 5 characters for counting the number of books, which allows 31^5 (28629151) books to be expanded. Then we have 3 digits left to encode the publication years and the item collections. After filtering the data, we find that the range of the publication year is no more than 70 (1951-2018), so 2 digits with 31^2 (961) distinct combinations is more than enough to store the information. As the starting year is given from 1600 in the program, so the publication year of the books can expand its range from 1600 to over 2500 year, allowing a great flexibility for the library to store the book from ancient times to the future. For the item collection, there are only 24 varieties in the data, so the rest 1 digit providing 31 unique combinations more sufficient enough to represent all the item collections. Besides, 7 extra collections can be expanded into the book identifier system in the future.

Considering the range of distinct number in each part we discussed above, we finally designated 5 characters for counting the number of books, 2 characters for counting the publication year and 1 for counting the item collections in the program. Hence, we acquire a 5-character code for number counting, 2-character code for publication year counting and 1-character code for item collection counting by converting those decimal numbers to a base-31 code.

2.4 Summary

Finally, we concatenate these 3 part of codes into an 8-character key for each book. In this case, we successfully create totally 250815 unique keys in the library system, except those with an unclear publication year (such as no publication year is shown in the data).

2.5 The Conversion System

Throughout our encoding process for the identifier of each book, we mainly resort to the method of conversion of number systems, which means we convert data of the books into a base-31 system. There are several advantages to use this number system conversion method.

First of all, the base-31 system helps us to generate a unique book counting number, a unique number to represent a year, and a unique number to represent the Item Collection. Then they together form a unique book identifier number which meet the project requirements.

Secondly, the characters in our base-31 system are clearly listed in the beginning and thus we avoid any possibility that might cause ambiguity in characters like 0 and O, v and u. Any characters among these in the original data are converted into characteristically distinct ones through our conversion system.

Thirdly, the base-31 system ensures a wide range of data space that it can record 31^5 books, year ranging from 1600 to 1600 + 31^2, in 31 Item Collections. So the same digits as in decimal number can represent much more possibilities in base-31 system.

3. Book Inquiry System

After successfully creating unique identifiers for each book, we constructed a book inquiry system to enable users and librarians to access the books through their identifiers, which allows us not only to search the book but also to verify the functionality as well as the validity of our identification system. By running several tests, we confirmed that the system was able to record the information and to give a unique identifier to each book. The searching process also reported the matching book given its identifier without case sensitivity. Therefore we verified the mathematical correctness of our identifier system.