Algorithms in Bioinformatics

Spring 2018

Hands on practices

Week 4: Sorting algorithms in different programming languages

Sorting is one of the most frequently used algorithms in almost all fields including Bioinformatics. You will see sorting is very import in many scenarios, especially when comparing large size data sets. In this week, we will finish at least two sorting algorithms in different languages and comparing their efficiency when they are applied to multiple data sets with different sizes.

All files and programs can be accessed under directory /share/home/ccwei/courses/2018/pab/week4/. A tar ball has been created for you to download from our course website and you can download or copy the tar ball to your own directory and extract the file with the command line as follows.

tar –xzvf week4\_algorithm\_language\_comparison.tarz

1. Datasets. In this lab, we have five datasets data1.list, data2.list, data3.list, data4.list and data5.list with the sizes of approximately 1, 10, 1,000, 10,000 and 1,000,000 numbers respectively. The sorted datasets were also provided together with the original datasets under the same directory:

data/

2. Programs. We have created three groups of programs in this lab.

2.1) insertion sort or insertion sort with binary search with time complexity of O(N^2)); merge sort with time complexity of O(NlogN). These two algorithms were implemented in three different programming languages: C, Python and Perl. All versions of the programs are listed under the directory.

sort/

You can get the usage of the programs by running them without any input.

For example, for the C program under

sort/C,

You can go to the directory and compile the .c file to generate an executable file with

gcc –o sort sort

Then you can get the usage of the program by running

./sort

without any input.

2.2) We have created a program to generate test datasets. All datasets included in this lab were generated by this program. The program is located at the directory

data/gen\_data.c

2.3) A comparing program to check the sorting result.

sort/diff\_data.c

3. You are asked to write the insertion sort or insertion sort with binary search and a merge sort program in a language of your own choice, and test your programs on the five datasets. You need to record the running time and memory spaces consumed. You can use shell command “time” to show the time taken for each program on a given dataset. You can use top to check the memory size of your program. If the program finish very quick, it is possible that you may not able to get the memory size (and it is ok if this happens).

You need to fill the following table and return it to teacher after the class. If you chose a language other than C, Python and Perl, please add two new columns to the table. You may not be able to finish testing the insertion sort for Data5. Please estimate the run time required to finish it by compute the theoretical time needed to finish the run.

Name: Student ID:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data | Size |  | C | | Python | | Perl | |  | |
|  |  |  | IS\* | M\*\* | IS | M | IS | M | IS | M |
| Data1 |  | Time (s) |  |  |  |  |  |  |  |  |
| Memory(MB) |  |  |  |  |  |  |  |  |
| Data2 |  | T(s) |  |  |  |  |  |  |  |  |
| M(MB) |  |  |  |  |  |  |  |  |
| Data3 |  | T(s) |  |  |  |  |  |  |  |  |
| M(MB) |  |  |  |  |  |  |  |  |
| Data4 |  | T(s) |  |  |  |  |  |  |  |  |
| M(MB) |  |  |  |  |  |  |  |  |
| Data5 |  | T(s) |  |  |  |  |  |  |  |  |
| M(MB) |  |  |  |  |  |  |  |  |

\*IS: Insertion Sort

\*\*M: Merge sort

Please summarize the above results in a report. You can put figures, tables or anything that you feel good to show what you have learned from this experiment. The report should be submitted together with homework2 before the class of April 2nd, 2018.