

Data Structures and Programming

Spring 2017, Programming # 1

The purpose of this assignment is to evaluate three on-line heuristic algorithms, namely, *move-to-front*, *transpose*, and *bit* on singly linked lists. The problem is as follows. Given a list L consisting of n distinct elements x_1, x_2, \dots, x_n , and a sequence of requests y_1, y_2, \dots, y_k (where $y_1, y_2, \dots, y_k \in \{x_1, x_2, \dots, x_n\}$), the problem is to design a singly linked list data structure for x_1, x_2, \dots, x_n so as to minimize the number of comparisons. (For example, in Fig.1 it takes 4 time units to find key 8.) It can be shown that in an off-line environment (i.e., y_1, y_2, \dots, y_k are given beforehand), the optimal strategy is to arrange x_1, x_2, \dots, x_n in a descending order according to their frequencies of occurrences in the request sequence. Now suppose requests y_1, y_2, \dots, y_k are given one at a time (i.e., in an on-line fashion). Move-to-front and transpose are two on-line algorithms that have been proposed in the literature to cope with the above on-line version of the list access problem. (See Figs 2 and 3 for example.) For Algorithm Bit, each item in the list maintains a bit that is complemented whenever the item is accessed. If an access causes a bit to change to 1, then the requested item is moved to the front of the list; otherwise the list remains unchanged. The bits of the items are initialized independently and uniformly at random.

In this assignment, L is assumed to be $\{0, 1, \dots, 9\}$. Write a program to perform the following:

- (1) Consider a sequence of requests of length 1000 (roughly) given on the webpage.
- (2) Calculate the cost of the optimal off-line algorithm described above. (To this end, first you have to find the the frequency of occurrences for each key in the generated request sequence.)
- (3) Assuming that L is $0 \rightarrow 1 \rightarrow \dots \rightarrow 9$ initially, calculate the cost with respect to the request sequence generated in Step 1 under the move-to-front heuristic.
- (4) Repeat Step 3 for the transpose heuristic.
- (5) Repeat Step 3 for the bit heuristic. To this end, use a random number generator to generate a sequence of ten 0 and 1 as the initial bit values for L .
- (6) Repeat Step 5 again. Note that here the random sequence is most likely to be different from the previous random sequence.
- (7) Output the following table:

Cumulated # Comparisons	Optimal	Move-to-Front	Transpose	Bit-1	Bit-2
$i = 50$					
$i = 100$					
$i = 150$					
...					
$i = 1000$					

in the following format:

```

Optimal      (50, n1)(100, n2)(150, n3)...(1000, n20)
Move-to-front (50, m1)(100, m2)(150, m3)...(1000, m20)
Transpose    (50, t1)(100, t2)(150, t3)...(1000, t20)
Bit-1       (50, b1)(100, b2)(150, b3)...(1000, b20)
Bit-2       (50, c1)(100, c2)(150, c3)...(1000, c20)
  
```

Note that those $n_i, m_i, t_i, b_i, c_i, 1 \leq i \leq 20$, are cumulated # of comparisons as explained above.

NOTE:

You need to document each of the subroutines in detail. In addition, your report must contain the following:

- (1) program files (source, object ...). Please name your executable file as "hw1.out".
 - (2) output: in the format explained above. Your program should read from file "request.txt". We will execute your program in the following way: ./hw1.out request.txt
 - (3) a "readme" file describing the language (C, C++, JAVA, ...), the environment (Windows, Linux, FreeBSD, Mac, ...), as well as the type of compiler needed to execute your program. (The TA will execute your codes; so be sure that your program is well-documented and executable.) Be sure to include your name, student ID No, and phone number (home and/or cell phone) in the "readme" file in case we need to reach you.
- Place all of the above in a zip file with your student ID and the version number as the file name. (For instance, if your ID is B89900000, and the file to be uploaded is version 1, your file name should be "B89900000_1.zip".)
 - Upload the zip file using "sftp" to 140.112.42.22:3922 with
 - username: ds17
 - password: datastructure

Find the /hw1/ directory and place your file in the directory. Note that those files already uploaded cannot be removed or altered. If you plan to upload a new version again, please increase the version number in your file name.

The "sftp" is readily available using "Filezilla" (supported by MS windows, linux, and osx), or simply use the "scp" command in unix.

NOTE: Should you have questions about the above, please send your inquiries to your TA at r05921040@ntu.edu.tw

The work must be the result of your individual effort. **No collaboration!!!**

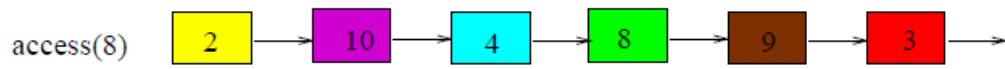


Fig. 1

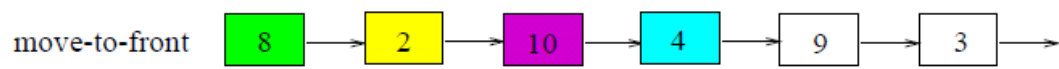


Fig. 2

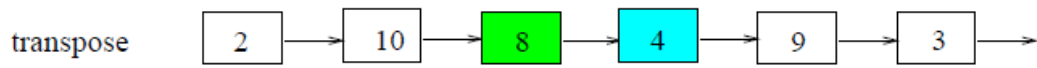


Fig. 3