REPORT

CptS 223 Project #1

Xiyu Xie

A: Problem statement.

In a circle (vector or list) of N person, starting from 0, delete the one after M person. Keep running until only one person left.

B: Algorithm design.

```
create_circle (circ, N, m)
    // create circ of N person, set the number of skipping to m, iter point at the first person
    for i<- 0 to N-1
        do      circ.push_back(Person(i))
    size<- N
    M<- m
    iter<- circ.begin()
elimateNext (circ, M, iter)
    // find the one should be deleted
        for i<- 1 to M
            do  iter++
            if  ( iter == circ.end())
            then iter<- circ.begin()
    // delete the one where iter pointing at, and set iter to the one next it
        iter<- circ.erase (iter)
        if  (iter == circ.end())
        then            iter<- circ.begin()
    // adjust size
        size--;
findWinner(circ, M, iter)
    while size>1
        do elimateNext
```

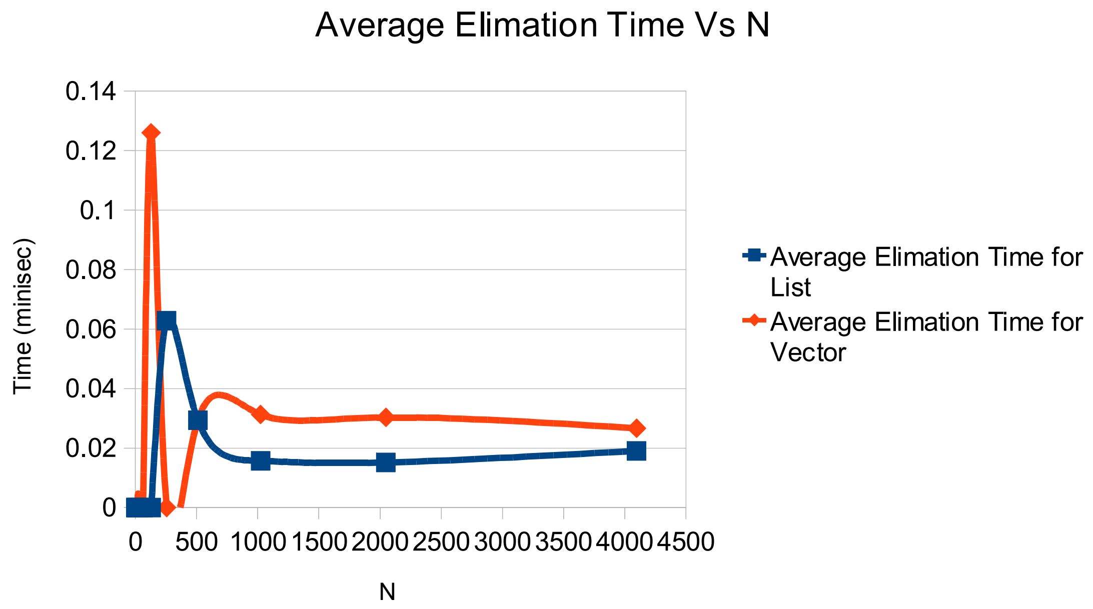
C: Experimental setup.

- Machine specification: windows 7, Visual Studio 2008,
- How many times did you repeat each experiment before reporting the final timing statistics?
  6 for vector, 2 for list

D: Experimental Results & Discussion.

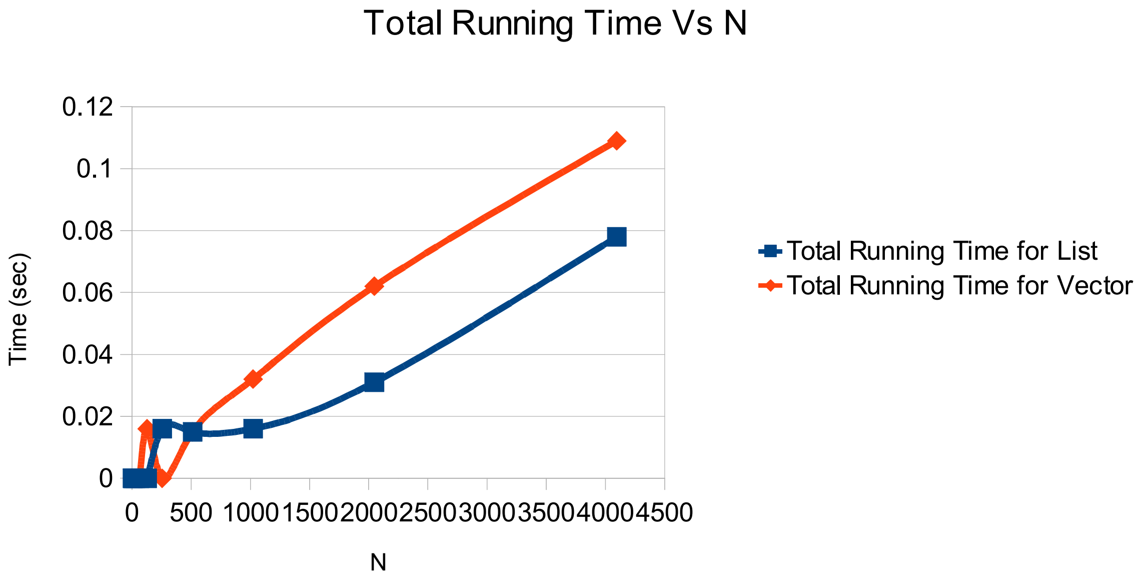
- Experiment #1: M= 3, N = 4,8,16,32, …, 10^12
- Experiment #2: N= 10^12, M = 2,4,8,... ,10^11 .

Plot II) average elimination time between on y-axis vs. N on x-axis;



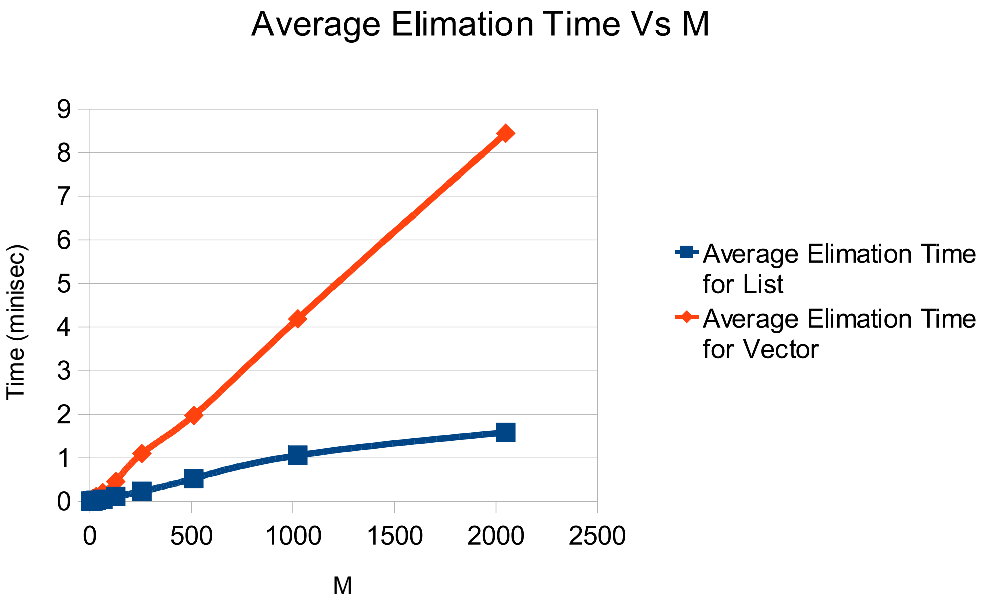
1. Vector takes more time then List, which is consistent with theoretical expectation. Because all the elements after the one being erased need to shift left.2.
2. Erasing time for both vector and list are almost constant, which **is contradict with my expectation.** I expect erase time would be Θ(N), as the average move for each erase are N/2. (which doesn't show after several repeat. And this result doesn't seem to be error, as N grows exponentially.)
   So I guess, elements shifting was not done every time after erasing. This shifting might be done when the relative size change reach a certain point

Plot I) total running time on y-axis vs. N on x-axis;



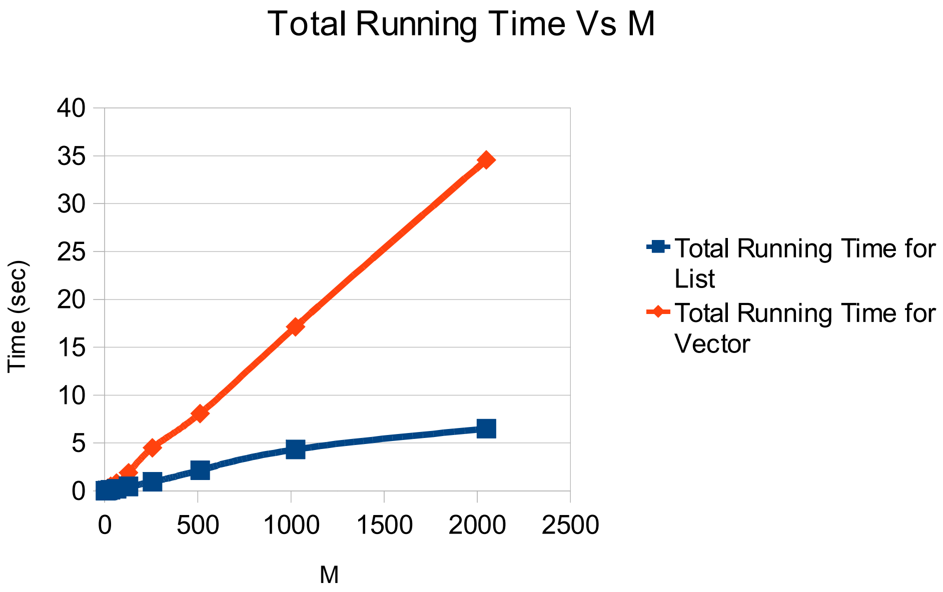
1. Vector takes more time then List, which is consistent with theoretical expectation, because of the shifting process.
2. Erasing time for both vector and list grow linearly with N size, which is consistent with theoretical expectation. Because every elimination is Θ(1), both experiment times N-1 times to find the winner. Thus, they are both Θ(N).

Plot IV) average elimination time on y-axis vs. M on x-axis;



1. Vector takes more time then List. Which is consistent with theoretical expectation, because of the shifting process.
2. Erasing time grow linearly for both container. Which is consistent with theoretical expectation. Because I use a for-loop to find the next one to be eliminated, which runs M times for each elimination.

Plot III) total running time on y-axis vs. M on x-axis;



1. Vector takes more time then list. Which is consistent with theoretical expectation, because of the shifting process.
2. Both container takes linear total running time relative to M. Which is consistent with theoretical expectation. As the average time is Θ(N), and both takes N-1, which is a constant to find the winner. So the total time is also Θ(N).

Summary:

In this experiment, list is a better container then vector in various cases of input N and M.