

Problem6: BinarySearch

Akira MATSUI

March 27, 2017

Introduction

In this exercise, we will learn binary search algorithm, which is to find a specific number in a sorted list. To use this search algorithm, you have to make sure that your list is ordered numerically. This exercise is related to bubble sort exercise.

If you have a number A that you need to find out where it is in a list, binary search algorithm tell you that in following steps. Suppose the list has N elements.

- compare A with the mean number R_m , of the list ($m := N/2$)
- if $A > R_m$, compare A with R_{m_2} ($m_2 := 3N/4$)
- if $A < R_m$, compare A with R_{m_2} ($m_2 := N/4$)

Here is the simple explanation.

There is a list that has numbers. Suppose that you want to find where $T = 9$ is, but cannot find it at a glance. This might happens if you list has a hundreds of elements.

(1,2,3,6,7,8,9)

First Step

Compare T with 6. In this case, $T > 6$ then go to next step.

Divide the list into half,

(1,2,3), (7,8,9).

Because $Target > 6$, we are interested in the latter list (7,8,9).

Second Step

You have

(7,8,9).

Compare $Target$ with 8. In this case, $Target > 8$ then we have found where $Target = 9$ is in the list.

Question

- you have `ToBeSorted <- sort(as.integer(runif(100, min = 1, max = 99)))`
- Implement `binary search` find where T is in `ToBeSorted`
- In this excersise, set `T <- as.integer(runif(1, min = 1, max = 99).`
- In the case `ToBeSorted` dose not have the same number as T , `print('can't find')`

Sample Answer

```
binary <- function(ToBeSorted,Target){

  N <- length(ToBeSorted) #Number of factor
  Start <- 1
  End <- N

  while(Start<End){
    Mid <- as.integer((Start + End)/2)
    if(ToBeSorted[Mid] == Target){
      return(Mid)
    }else if(Target < ToBeSorted[Mid]){
      End <- Mid
    }else{
      Start <- Mid + 1
    }
  }
  print('Not found')
}

ToBeSorted <- sort(as.integer( runif(100, min = 1, max = 99) ))
Target <- as.integer( runif(1, min = 1, max = 99) )

print(Target)

## [1] 91

ToBeSorted[binary(ToBeSorted,Target)]

## [1] "Not found"

## [1] NA

print(ToBeSorted)

## [1] 2 2 4 5 6 6 7 8 8 9 9 10 10 10 11 12 12 13 14 15 17 18 19
## [24] 19 20 22 24 25 26 27 27 29 30 30 30 30 31 32 32 33 35 36 41 42 42 43
## [47] 44 46 49 50 52 55 56 56 59 59 59 60 63 63 64 64 64 65 68 70 70 71 71
## [70] 71 72 72 72 73 73 75 75 75 76 76 78 79 79 80 81 81 82 84 84 84 85 86
## [93] 88 90 92 93 94 96 96 98
```