Lab9

2018/11/13

Q1

```
set.seed(999)
library(microbenchmark)

odd_count<-function (x){
  odd_num = 0
  for (i in 1:length(x)){
    if (x[i] %% 2 ==1)
      odd_num <- odd_num +1
  }
  return(odd_num)
}

odd_count2 <- function(x){
  odd_num = 0
  odd_num <- sum(x %% 2 == 1)
  return(odd_num)
}

microbenchmark(odd_count(1:1000), odd_count2(1:1000))
```

```
## Unit: microseconds
##                 expr     min      lq      mean  median       uq       max
##   odd_count(1:1000) 178.470 219.449 849.80123 410.436 525.2315 36138.614
##  odd_count2(1:1000)   9.629  17.027  63.63424  24.087  31.6425  3757.151
##  neval
##    100
##    100
```

We found that the Vectorized one is faster.

Q2

```
sort_vec<-function (x, ascending = TRUE){
  if(length(x) <2){
    return (x)
  }
  else{
    for (last in length(x):2){
      for (first in 1:(last -1)){
        if (x[first]> x[first+1]){
          temp <-x[first]
          x[first] = x[first +1]
          x[first +1] = temp
        }
      }
```

```
    }
  }
  if (ascending == FALSE){
    for (last in length(x):2){
      for (first in 1:(last-1)){
        if(x[first]<x[first +1]){
          temp<-x[first]
          x[first] = x[first +1]
          x[first +1] = temp
        }
      }
    }
  }
  return (x)
}

sort_vec(c(3, 1, 2), ascending = TRUE)
```

```
## [1] 1 2 3
```

```
sort_vec(c(3, 1, 2), ascending = FALSE)
```

```
## [1] 3 2 1
```

Q3

N=1000

```
set.seed(999)
N = 1000
#dynamically allocated memory
data_series = 0
system.time({for (i in 2:N){
  data_series[i] = data_series[i-1] + sample(c(-1, 1), 1)
  }
})
```

```
##    user  system elapsed
##   0.007   0.000   0.008
```

```
#preallocated memory
system.time({
  data_series2 = vector("numeric", N)
  for (i in 2:N) {
    data_series2[i] = data_series2[i-1] + sample(c(-1, 1), 1)
  }
})
```

```
##    user  system elapsed
##   0.006   0.000   0.006
```

There is not obvious difference between dynamically allocated memory and preallocated memory

N=10000

```
set.seed(999)
N = 10000
```

```
#dynamically allocated memory
data_series = 0
system.time({for (i in 2:N){
  data_series[i] = data_series[i-1] + sample(c(-1, 1), 1)
}
})
```

```
##    user  system elapsed
##   0.052   0.010   0.062
```

```
#preallocated memory
system.time({
  data_series2 = vector("numeric", N)
  for (i in 2:N) {
    data_series2[i] = data_series2[i-1] + sample(c(-1, 1), 1)
  }
})
```

```
##    user  system elapsed
##   0.042   0.005   0.048
```

When N = 10000, we find that the preallocated memory is faster than the dynamically allocated memory but there only a tiny difference.

N=1000000

```
set.seed(999)
N = 1000000
#dynamically allocated memory
data_series = 0
system.time({
  for (i in 2:N){
  data_series[i] = data_series[i-1] + sample(c(-1, 1), 1)
}
})
```

```
##    user  system elapsed
##   3.836   0.409   4.254
```

```
#preallocated memory
system.time({
  data_series2 = vector("numeric", N)
  for (i in 2:N) {
    data_series2[i] = data_series2[i-1] + sample(c(-1, 1), 1)
  }
})
```

```
##    user  system elapsed
##   3.566   0.358   3.929
```

When N = 1000000, we find that there is obvious difference between dynamically allocated memory and preallocated memory that the second one is much faster than the first one.