Acceleration

目录

```
library(Matrix)
```

```
## Warning: 程辑包'Matrix'是用R版本4.3.1 来建造的
```

```
rm(list = ls())
iter <- 10
time <- 0

for (i in 1:iter) {
  c <- system.time({n <- 2500
  A <- matrix(runif(n * n), nrow = n, ncol = n)
  trans_A <- t(A)
  A_Square <- A * A})[3]
  time <- time + c
}

time_1 <- time / iter
cat(" 任务一平均用时:", time_1, " 秒.\n")
```

```
## 任务一平均用时: 0.135 秒.
```

```
iter <- 10
time <- 0
n <- 2400
N <- 1000
for (i in 1:iter) {
  c <- system.time({A <- matrix(rnorm(n * n), nrow = n, ncol = n)
  A_N <- A ^ N})[3]
  time <- time + c
}

time_2 <- time / iter
cat(" 任务二平均用时:", time_2, " 秒.\n")
```

```
## 任务二平均用时: 0.75 秒.
```

```
iter <- 10
time <- 0
n <- 7e6

for (i in 1:iter) {
  c <- system.time({randoms <- rnorm(n)
  sorted <- sort(randoms)})[3]
  time <- time + c
}

time_3 <- time / iter
cat(" 任务三平均用时:", time_3, " 秒.\n")
```

```
## 任务三平均用时: 0.563 秒.
```

```
rm(list = ls())
iter <- 10
time <- 0
n <- 2800

for (i in 1:iter) {
  c <- system.time({a <- rnorm(n)
  b <- crossprod(a)})[3]
  time <- time + c
}

time_4 <- time / iter
cat(" 任务四平均用时:", time_4, " 秒.\n")
```

```
## 任务四平均用时: 0 秒.
```

```
iter <- 10
time <- 0
n <- 3000
p <- 10

for (i in 1:iter) {
  c <- system.time({X <- matrix(rnorm(n * p), nrow = n, ncol = p)
  Y <- rnorm(n)
  beta <- solve(crossprod(X)) %*% t(X) %*% Y})[3]
```

```
  time <- time + c
}

time_5 <- time / iter
cat(" 任务五平均用时:", time_5, " 秒.\n")
```

```
## 任务五平均用时: 0.002 秒.
```

```
iter <- 10
time <- 0
n <- 2e6

for (i in 1:iter) {
  c <- system.time({randoms <- rnorm(n)
  result <- fft(randoms)})[3]
  time <- time + c
}

time_6 <- time / iter
cat(" 任务六平均用时:", time_6, " 秒.\n")
```

```
## 任务六平均用时: 0.123 秒.
```

```
iter <- 10
time <- 0
n <- 640

for (i in 1:iter) {
  c <- system.time({matrix <- matrix(rnorm(n * n), nrow = n, ncol = n)
  eigenvalues <- eigen(matrix)$values})[3]
  time <- time + c
}

time_7 <- time / iter
cat(" 任务七平均用时:", time_7, " 秒.\n")
```

```
## 任务七平均用时: 0.358 秒.
```

```
rm(list = ls())
iter <- 10
time <- 0
```

```r
n <- 2500

for (i in 1:iter) {
  c <- system.time({matrix <- matrix(rnorm(n * n), nrow = n, ncol = n)
  det_value <- det(matrix)})[3]
  time <- time + c
}

time_8 <- time / iter
cat(" 任务八平均用时:", time_8, " 秒.\n")
```

```
## 任务八平均用时: 0.285 秒.
```

```r
rm(list = ls())
iter <- 10
time <- 0
n <- 3000

for (i in 1:iter) {
  c <- system.time({matrix <- matrix(rnorm(n * n), nrow = n, ncol = n)
  sym_matrix <- t(matrix) %*% matrix
  cholesky_matrix <- chol(sym_matrix, lower = TRUE)})[3]
  time <- time + c
}

time_9 <- time / iter
cat(" 任务九平均用时:", time_9, " 秒.\n")
```

```
## 任务九平均用时: 0.656 秒.
```

```r
iter <- 10
time <- 0
n <- 1600

for (i in 1:iter) {
  c <- system.time({matrix <- matrix(rnorm(n * n), nrow = n, ncol = n)
  solve(matrix)})[3]
  time <- time + c
}

time_10 <- time / iter
cat(" 任务十平均用时:", time_10, " 秒.\n")
```

```
## 任务十平均用时: 0.16 秒.
```

```
iter <- 10
time <- 0
n <- 3.5e6

for (i in 1:iter) {
  c <- system.time({fibonacci_numbers <- numeric(n)
  fibonacci_numbers[1:2] <- c(0, 1)
  for (j in 3:n) {
    fibonacci_numbers[j] <- fibonacci_numbers[j - 1] + fibonacci_numbers[j - 2]
  }})[3]
  time <- time + c
}

time_11 <- time / iter
cat(" 任务十一平均用时:", time_11, " 秒.\n")
```

```
## 任务十一平均用时: 0.228 秒.
```

```
rm(list = ls())
iter <- 10
time <- 0
n <- 3000

for (i in 1:iter) {
  c <- system.time({hilbert_matrix <- Hilbert(n)})[3]
  time <- time + c
}

time_12 <- time / iter
cat(" 任务十二平均用时:", time_12, " 秒.\n")
```

```
## 任务十二平均用时: 0.089 秒.
```

```
rm(list = ls())
gcd_recursive <- function(a, b){
    if (b == 0) return(a)
    else{
      return(gcd_recursive(b, a%%b))
    }
}
```

```r
iter <- 10
time <- 0
n <- 400000
p <- 1000

for (i in 1:iter) {
  c <- system.time({pairs_num <- n
  pairs <- matrix(sample(1:p, pairs_num * 2, replace = TRUE), ncol = 2)
  gcd_values <- numeric(pairs_num)
  for (j in 1:pairs_num) {
    gcd_values[j] <- gcd_recursive(pairs[j, 1], pairs[j, 2])
  }})[3]
  time <- time + c
}
time_13 <- time / iter
cat(" 任务十三平均用时:", time_13, " 秒.\n")
```

```
## 任务十三平均用时: 1.239 秒.
```

```r
rm(list = ls())
iter <- 10
time <- 0
n <- 500

for (k in 1:iter) {
  c <- system.time({first_row <- 1:n
  toeplitz_matrix <- matrix(0, nrow = n, ncol = n)
  for (i in 1:n) {
    for (j in 1:n) {
      toeplitz_matrix[i, j] <- first_row[abs(i - j) + 1]
    }
  }})[3]
  time <- time + c
}

time_14 <- time / iter
cat(" 任务十四平均用时:", time_14, " 秒.\n")
```

```
## 任务十四平均用时: 0.024 秒.
```

```r
rm(list = ls())
iter <- 10
time <- 0
n <- 100
p <- 45

for (k in 1:iter) {
  c <- system.time({data_matrix <- matrix(rnorm(n * p), nrow = n, ncol = p)
  pca_result <- prcomp(data_matrix)})[3]
  time <- time + c
}

time_15 <- time / iter
cat(" 任务十五平均用时:", time_15, " 秒.\n")
```

```
## 任务十五平均用时: 0.003 秒.
```