1er Parcial Análisis Numérico

Federico Torres Mojica

23 de agosto de 2019

1 Primer Punto

1. Sea f(n) la eficiencia del algoritmo, medida como el número mínimo de operaciones requeridas para resolver el problema.

a) Implemente en R o Python un algoritmo que le permita sumar únicamente los elementos de la sub matriz triangular superior o triangular inferior, dada la matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.
b) Implemente en R o Python un algoritmo que le permita sumar los elementos de una matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.
c) Implemente en R o Python un algoritmo que le permita sumar los n2 primeros números naturales al cuadrado. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.

Solución:
Lo primero que se realiza en la implementación del algoritmo es crear la matriz cuadrada An a partir de un numero n aleatorio que determina su tamaño y luego se geenera una secuencia de números aleatorios que serán las entradas de la matriz An. Posteriormente se hace uso de dos funciones propias del lenguaje R para transformar la matriz inicial An en una matriz triangular inferior cuyas entradas en la submatriz triangular superior y en su diagonal sean iguales a cero, de tal forma que al sumar todos los elementos de la matriz solo se tengan en cuenta las entradas de las submatriz triagular inferior. El recorrido de la matriz se logra mediante un ciclo iterativo, de tal forma que al finalizar el proceso de ejecución se habrán hecho un total de nxn operaciones y por lo tanto la eficiencia del algoritmo sería:

$$f(n) = n^2 \tag{1}$$

Al final del documento se muestran los resultados obteidos del algoritmo para matrices An con n = 7, n = 8 y n = 9.

En el apartado b se pide sumar todos los elementos de una matriz cuadrada nxn por lo que el proceso es muy similar al anterior pero sin necesidad de transformar la matriz inicial. Por lo tanto, en este caso la eficiencia del algoritmo f(n) nuevamente es de nxn, o lo que es lo mismo

$$f(n) = n^2 \tag{2}$$

Al final del documento se muestran los resultados obteidos del algoritmo para matrices An con n = 3, n = 4 y n = 6.

Para sumar los n2 primeros numeros naturales al cuadrado, se procedió a generar un numero n aleatorio mediante la función sample() de R y luego se genera una secuencia ascendente de numeros desde 1 hasta el numero n mediante la funcion seq() en R. A esta secuencia de numeros se aplica un ciclo iterativo que la recorra y que para cada caso los multiplique por si mismos y luego los sume al contador suma inicializado en cero. De esta forma, para cada numero el algoritmo realiza dos operaciones y su complejidad expresada en terminos de n sería:

Al final del documento se muestran los resultados obteidos del algoritmo para matrices An con n = 3, n = 9 y n = 5.

$$f(n) = 2n \tag{3}$$

2 Segundo Punto

En R: Sean f(x) = ln(x + 2) y g(x) = sin(x) dos funciones de valor real. a) Utilice la siguiente formula recursiva con E = 10(-8) para hallar el punto de intersección:

Ver figura 8.

Solución: Se procedió a implementar un algoritmo que haga uso de la ecuación suministrada para iterar sobre el rango del dominio de x donde se produce la intersección. El primer paso es graficar las dos ecuaciones iniciales para verificar manualmente cual es el intervalo óptimo inicial sobre el cual aplicar la ecuación de convergencia.

Posteriormente se van calculando iterativamente los puntos x0 y x1 hasta que el error resultante de convergencia sea menor a E = 10(-8). En este caso, x0 = -1.8 y x1 = -1.1 y el resultado de la ejecución arroja que el punto de intersección de las dos ecuaciones iniciales es 6.283185 con un total de 7 iteraciones.

```
> #Punto 1
> #a
> n = sample(1:10,1); n
[1] 7

> sample(1:20,n,replace = TRUE)
[1]  4 17  1  4  7  7  8

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 14.793739 16.783212  4.843342 13.002567  8.251276  4.207355  5.709217
[2,]  9.387838  3.732303  9.932981 20.407095 19.446154 10.604054 19.676500
[3,] 13.359561 20.287726  5.913023  3.052322 16.881326  4.458836  5.992549
[4,] 12.894468  4.931226 20.017951 11.165478  8.785694 12.859901  8.196341
[5,] 17.489307 15.521387  5.366006  9.288799 13.747009 17.100676 20.169016
[6,]  9.483718 10.096697 19.088144 10.981972 10.261191 11.253324  8.942761
[7,] 18.936403 13.222788  4.608009  3.486338 12.608938  9.098218 16.453055

> LA = A

> LA[col(LA) >= row(LA)] = 0; LA
          [,1]      [,2]      [,3]      [,4]     [,5]     [,6] [,7]
[1,]  0.000000  0.000000  0.000000  0.000000  0.00000 0.000000    0
[2,]  9.387838  0.000000  0.000000  0.000000  0.00000 0.000000    0
[3,] 13.359561 20.287726  0.000000  0.000000  0.00000 0.000000    0
[4,] 12.894468  4.931226 20.017951  0.000000  0.00000 0.000000    0
[5,] 17.489307 15.521387  5.366006  9.288799  0.00000 0.000000    0
[6,]  9.483718 10.096697 19.088144 10.981972 10.26119 0.000000    0
[7,] 18.936403 13.222788  4.608009  3.486338 12.60894 9.098218    0

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 566.7778
> 
```

Figure 1: Resultado para n = 7

```
> #Punto 1
> #a
> n = sample(1:10,1); n
[1] 8

> sample(1:20,n,replace = TRUE)
[1] 17  2 17  3 16 20 10 20

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
[1,]  3.771977 19.141672  4.098484 20.748972  5.334670  3.178808  7.958565  3.370971
[2,] 16.823150 15.105326 20.581284 13.285598  4.968448  4.496638  9.071875  7.769801
[3,] 18.260291 20.066123  4.092443  3.368887 11.687032 17.495451 19.059430 10.349019
[4,]  3.598805 18.899645  4.326176 10.409659 18.835759 17.414403  6.836446 16.978141
[5,] 15.602748  9.932179 16.351253 10.467849 17.255290 18.225137 14.182266  6.993120
[6,] 14.391746  8.427310 14.416514 19.498134 10.488961 19.249554 15.455503  3.898891
[7,] 18.868519 18.379806 18.805918  3.511143 16.627075 15.874133 12.096615  9.557894
[8,] 17.178900 14.217958  6.173956  6.514355 19.737851  3.758290  9.198390 16.487726

> LA = A

> LA[col(LA) >= row(LA)] = 0; LA
          [,1]      [,2]      [,3]      [,4]     [,5]     [,6]    [,7] [,8]
[1,]  0.000000  0.000000  0.000000  0.000000  0.00000  0.00000 0.00000    0
[2,] 16.823150  0.000000  0.000000  0.000000  0.00000  0.00000 0.00000    0
[3,] 18.260291 20.066123  0.000000  0.000000  0.00000  0.00000 0.00000    0
[4,]  3.598805 18.899645  4.326176  0.000000  0.00000  0.00000 0.00000    0
[5,] 15.602748  9.932179 16.351253 10.467849  0.00000  0.00000 0.00000    0
[6,] 14.391746  8.427310 14.416514 19.498134 10.48896  0.00000 0.00000    0
[7,] 18.868519 18.379806 18.805918  3.511143 16.62707 15.87413 0.00000    0
[8,] 17.178900 14.217958  6.173956  6.514355 19.73785  3.75829 9.19839    0

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 783.2089
```

Figure 2: Resultado para n = 8

```
> #Punto 1
> #a
> n = sample(1:10,1); n
[1] 9

> sample(1:20,n,replace = TRUE)
[1]  2 11  1 12 15  6  5  1  1

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
 [1,] 10.619715  5.239376  6.350830 18.250806 14.756804  8.110221  4.393933  9.265441 15.847920
 [2,] 11.773998  3.293788  9.333514  4.113082 15.085083 19.295974 13.343705 17.849719 13.808548
 [3,]  8.219808  7.191183  8.285076  4.440945  4.855258  9.141337  7.077337 16.245841  9.953981
 [4,] 16.005113 17.374985  5.103701 12.765565 13.301311  4.878779  5.225727  5.475844  4.324129
 [5,]  9.837447 19.106138 12.243493  5.062698 17.218375 12.207096 10.654028 11.237029  9.769249
 [6,] 11.269355 12.344257  7.463008 18.067310 16.110579 20.242064 16.766584 12.140066  6.386244
 [7,] 14.693753 18.050542 13.849313  4.293905 17.457131  8.555107 13.369671 20.855089 19.791811
 [8,] 12.006185 14.217318 14.108617  6.812092  5.607038 13.530463  5.121882  4.289481  5.927365
 [9,] 17.441059 13.073826 19.332330 12.682604 20.192311 15.093011 14.052185  7.281755  3.195173

> LA = A

> LA[col(LA) >= row(LA)] = 0; LA
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]     [,8] [,9]
 [1,]  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000    0
 [2,] 11.773998  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000    0
 [3,]  8.219808  7.191183  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000    0
 [4,] 16.005113 17.374985  5.103701  0.000000  0.000000  0.000000  0.000000 0.000000    0
 [5,]  9.837447 19.106138 12.243493  5.062698  0.000000  0.000000  0.000000 0.000000    0
 [6,] 11.269355 12.344257  7.463008 18.067310 16.110579  0.000000  0.000000 0.000000    0
 [7,] 14.693753 18.050542 13.849313  4.293905 17.457131  8.555107  0.000000 0.000000    0
 [8,] 12.006185 14.217318 14.108617  6.812092  5.607038 13.530463  5.121882 0.000000    0
 [9,] 17.441059 13.073826 19.332330 12.682604 20.192311 15.093011 14.052185 7.281755    0

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 923.6044
> |
```

Figure 3: Resultado para n = 9

```
> #Punto 1
> #b
> n = sample(1:10,1); n
[1] 3

> sample(1:20,n,replace = TRUE)
[1]  9 15  1

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
          [,1]     [,2]      [,3]
[1,] 18.753535 11.34980  3.460265
[2,]  9.285262 13.18937 11.073170
[3,]  5.153839 10.88928 19.244408

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 102.3989
> 
```

Figure 4: Resultado para n = 3

```
> #Punto 1
> #b
> n = sample(1:10,1); n
[1] 4

> sample(1:20,n,replace = TRUE)
[1] 11 11 13 19

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
         [,1]      [,2]      [,3]      [,4]
[1,] 12.49807 19.821153  3.956092 20.999966
[2,] 20.37832  5.209906 12.942353 14.279065
[3,] 11.95302 10.659901 16.491723  9.479627
[4,]  3.29510  8.832256 13.961603  3.419988

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 188.1781
> 
```

Figure 5: Resultado para n = 4

```
> #Punto 1
> #b
> n = sample(1:10,1); n
[1] 6

> sample(1:20,n,replace = TRUE)
[1] 11  2 13  6  8 16

> A =matrix(c(runif(n*n, min=3, max=21)), nrow = n, ncol = n); A
          [,1]      [,2]      [,3]     [,4]      [,5]      [,6]
[1,]  9.249808 13.452653  8.809837 12.17760  4.194617 10.731656
[2,] 10.502913  3.541131  8.172135 13.59219 10.975575 16.704748
[3,]  5.946215 15.835562 17.676898 14.55822 16.273129 14.466698
[4,] 11.489600 16.095433  4.266183 10.50424 12.726427  4.005698
[5,] 17.073453 12.711993  8.620779 11.85225  5.797331 12.616007
[6,] 15.713333  8.114883 15.301191 16.70907  9.037167  5.428437

> suma = 0

> for(i in A){
+   suma = suma + i
+ }; suma
[1] 404.9251
>
```

Figure 6: Resultado para n = 6

```
> #Punto 1
> #c
> n = sample(1:10,1); n
[1] 3

> numeros = seq(from = 0, to = n)

> suma = 0

> for(i in numeros){
+   suma = suma + i^2
+ }; suma
[1] 14
> source('~/FEDERICO/Parcial 1/Punto_3.R', echo=TRUE)

> #Punto 1
> #c
> n = sample(1:10,1); n
[1] 9

> numeros = seq(from = 0, to = n)

> suma = 0

> for(i in numeros){
+   suma = suma + i^2
+ }; suma
[1] 285
> source('~/FEDERICO/Parcial 1/Punto_3.R', echo=TRUE)

> #Punto 1
> #c
> n = sample(1:10,1); n
[1] 5

> numeros = seq(from = 0, to = n)

> suma = 0

> for(i in numeros){
+   suma = suma + i^2
+ }; suma
[1] 55
> |
```

Figure 7: Resultado para n = 3 , n = 9, n = 5

Figure 8: Ecuación

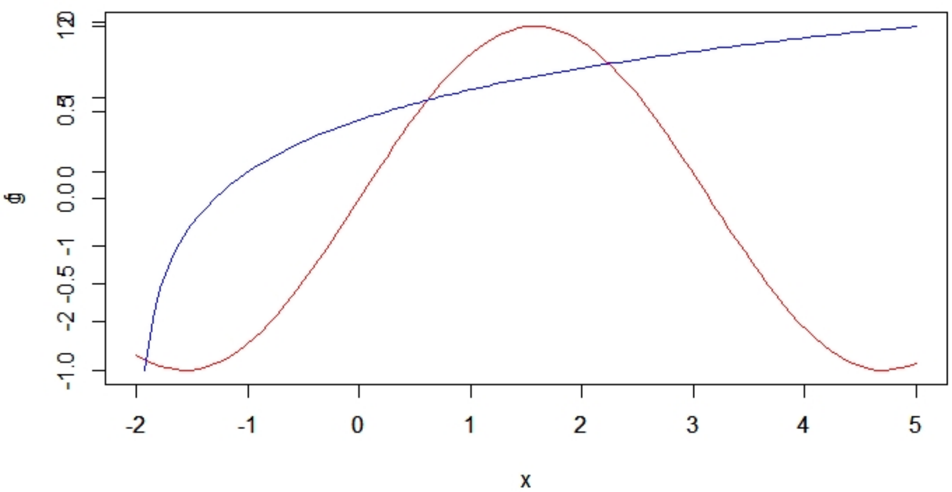

Figure 9: Gráfica