



5

Instituto Politécnico Nacional Escuela Superior de Computo

Materia

Profesor

Practica

Alumno:

- García González Aarón Antonio
 - Zúñiga Pineda Alvaro
- Arista Bojorges Victor Usiel

Agosto 26, 2019



Índice

Hoja de practica	¡Error! Marcador no definido.
Hoja de soluciones	¡Error! Marcador no definido.
.....	¡Error! Marcador no definido.
Código fuente	¡Error! Marcador no definido.
Impresiones de pantalla.....	¡Error! Marcador no definido.

Ejercicio 1

```
# Funcion que multiplica dos numeros
def multiply(x,y):
    if y == 0:
        return 0
    z = multiply(x,y/2)
    if not(y & 1):
        return 2*z
    else:
        return x + 2*z
```

a) Prueba de escritorio

```
MacBook-Pro-de-Aaron:p1 aarongarcia$ python index.py
Teclea el primer numero: 0
Teclea el segundo numero: 0
0
MacBook-Pro-de-Aaron:p1 aarongarcia$ python index.py
Teclea el primer numero: 100
Teclea el segundo numero: 8
800
MacBook-Pro-de-Aaron:p1 aarongarcia$ █
```

b) ¿Por qué es correcto el algoritmo?

Para empezar el algoritmo realiza lo que tiene que realizar (propósito) y lo hace correctamente

c) ¿Tiempo computacional? Es $O(\ln(y))$, donde en el peor de los casos es $O(y/2)$

Ejercicio 2

```
# Funcion que llena una matriz a partir de un arreglo, dado una matriz B[i][j],
# si i < j, entonces el valor en la posicion (i,j) en la matriz sera la suma de
# A[i] hasta A[j], de lo contrario, es decir i>j, el valor no esta definido.
int A[n][n];
for(i=0; i<n; i++){
    aux=a[i];
    for(j=0; j<n; j++){
        A[i][j]=0;
        if (i<j){
            aux=aux + a[j];
            A[i][j]=aux;
        }
        printf("%d\t", A[i][j]);
    }
    aux=0;
    printf ("\n");
}
```

- a) ¿Por qué el algoritmo es correcto?
Igualmente es correcto porque el algoritmo realiza todo lo que tiene que hacer con base a las entradas que se le dan al mismo, pero tampoco es el mas eficiente debido a que se podría optimizar reduciendo el numero de ciclos for que contiene, con la excepción que no se encontró dicha solución.
- b) Tiempo computacional del problema
Al parecer es $O(n^3)$
- c) Prueba de escritorio

```

ingrese el tamao de matriz que quiera: 5
introduce el valor que quieres: 1
introduce el valor que quieres: 2
introduce el valor que quieres: 3
introduce el valor que quieres: 4
introduce el valor que quieres: 5
0      3      6      10     15
0      0      5      9      14
0      0      0      7      12
0      0      0      0      9
0      0      0      0      0

```

Ejercicio 3

Funcion que ordena un arreglo mediante el metodo de seleccion, es decir buscar
buscar el menor de los numeros en cada iteracion y los coloca en el lugar correcto

```

def selection(A):
    for i in range(len(A)):
        minimo=i
        for j in range(i,len(A)):
            if(A[j] < A[minimo]):
                minimo=j
        if(minimo != i):
            aux=A[i]
            A[i]=A[minimo]
            A[minimo]=aux
    return A

```

- a) Prueba de escritorio

```

MacBook-Pro-de-Aaron:p1 aarongarcia$ python index.py
('Arreglo original ', [11, 3, 9, 9, 9, 24, 6, 18, 21, 5, 97])
('Arreglo ordenado ', [3, 5, 6, 9, 9, 9, 11, 18, 21, 24, 97])

```

- b) Tiempo computacional: $O(n^2)$

Ejercicio 4

```
void Algorithm1 (int a, int b, int *resul)
{
    int u,v,q,r,x1,x2,y1,y2,x,y, d;
    u=a;
    v=b;
    x1=1;
    y1=0;
    x2=0;
    y2=1;
    printf ("u\tv\tq\t r\t x1\t x2\t y1\t y2\t x\t y\n");
    while(u!=0)
    {
        q=v/u;
        r=v-(q*u);
        x=x2-(q*x1);
        y=y2-(q*y1);

        v=u;
        u=r;
        x2=x1;
        x1=x;
        y2=y1;
        y1=y;

        printf ("%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",u,v,q,r,x1,x2,y1,y2,x,y);
    }
    d=v;
    x=x2;
    y=y2;
    resul[0]=d;
    resul[1]=x;
    resul[2]=y;
}
```

```
Ingrese el valor de a: 5
Ingrese el valor de b: 15
u      v      q      r      x1      x2      y1      y2      x      y
0      5      3      0      -3      1      1      0      -3      1
5 1 0
```

- a) Complejidad: $O(\log(n))$
- b) ¿Qué hace? Encuentra el máximo común divisor, además la combinación lineal de los números ingresados para obtener el máximo común múltiplo

Ejercicio 5

```
def algorithm2(a,p):  
    u = a; v = p  
    x1 = 1; x2 = 0  
  
    q = 0  
    r = 0  
    x = 0  
    contador = 0  
  
    while u != 1:  
        q = v/u  
        r = v - (q*u)  
        x = x2 - (q*x1)  
        v = u  
        u = r  
        x2 = x1  
        x1 = x  
        print("Iterecacion: ", contador)  
        print("q", q)  
        print("v", v)  
        print("u", u)  
        print("x", x)  
        print("x1", x1)  
        print("x2", x2)  
        print("r", r)  
        print("p", p)  
        contador += 1  
  
    return x1 % p
```

```
MacBook-Pro-de-Aaron:p1 aarongarcia$ python index.py
('Iterecacion: ', 0)
('q', 1)
('v', 3)
('u', 2)
('x', -1)
('x1', -1)
('x2', 1)
('r', 2)
('p', 5)
('Iterecacion: ', 1)
('q', 1)
('v', 2)
('u', 1)
('x', 2)
('x1', 2)
('x2', -1)
('r', 1)
('p', 5)
2
```

- a) ¿Qué hace? Encuentra un numero que multiplicado por el primer numero y obteniendo el modulo con el segundo, da uno, o visto de otra manera que el primer numero exceda en una unidad a el múltiplo del segundo
- b) Complejidad: $O(\log(n))$