

# Estructuras de Datos 1 - ST0247

## Primer Parcial Grupo 033 (Lunes)

Nombre .....

Departamento de Informática y Sistemas

Universidad EAFIT

12 de marzo de 2018

### 1 Recursión 20%

El pequeño Polka fue a la tienda a comprar algunas cosas para su madre. El pequeño Polka pagó cierta cantidad de dinero y el tendero debe devolverle exactamente  $K$  pesos. El tendero tiene exactamente  $n$  diferentes tipos de monedas en su tienda, cada una de ellas con un valor  $v_i$ . Como el tendero no es muy astuto en asuntos de matemáticas, te ha pedido el favor de decirle de cuantas maneras puede él devolverle tal cantidad a Polka. Por favor ayúdanos a completar el siguiente código.

- **Ejemplo:** Sea  $n = |v| = 3, v = \{3, 4, 1\}, K = 7$ . Las formas posibles de devolver 7 dólares serían 5:

1.  $1 + 1 + 1 + 1 + 1 + 1 + 1$
2.  $3 + 1 + 1 + 1$
3.  $3 + 3 + 1$
4.  $3 + 4$
5.  $4 + 1 + 1 + 1$

```
1 int cuantas(int K, int[] v, int n){
2     if(K == 0){
3         return 1;
4     }
5     boolean imposible;
6     imposible = n <= 0 && K >= 1;
7     imposible = imposible || K < 0;
8     if(imposible){
9         -----
10    }
11    int ni = cuantas(K, v, n - 1);
12    int nj = cuantas(K - v[n-1], v, n);
13    int suma = -----;
14    return suma;
15 }
```

a (10%) Línea 9 .....

b (10%) Línea 13 .....

### 2 Complejidad 40%

- a (10%) Asuma que  $func(n)$  ejecuta  $T(n) = (n - 1)^2$  pasos. ¿Cuál de las siguientes afirmaciones es correcta?

```
1 void misterio(int n){
2     for(int i = 1; i * i <= n; i++){
3         int m = func(n);
4     }
5 }
```

- $misterio(n)$  ejecuta  $(n - 1)^2$  pasos.
- $misterio(n)$  ejecuta  $(n - 1)^3$  pasos.
- $misterio(n)$  ejecuta  $(n - 1)^4$  pasos.
- $misterio(n)$  ejecuta  $\sqrt{n}(n - 1)^2$  pasos.

- b (10%) ¿Cuál de las siguientes afirmaciones es correcta para la función  $func1(n)$ ?

```
1 void func1(int n){
2     for(int i = 0; i <= n; i++){
3         int ii = n / 5;
4         for(int j = 0; j <= n - ii * 5;
5             j++){
6             print(j);
7         }
8     }
```

- $func1(n)$  ejecuta  $O(\sqrt{n})$  pasos
- $func1(n)$  ejecuta  $O(n^2)$  pasos
- $func1(n)$  ejecuta  $O(n)$  pasos
- $func1(n)$  ejecuta  $O(n^3)$  pasos

- c (10%) ¿Cuál de las siguientes afirmaciones es correcta para la función  $func2(n, m)$ ?

```
1 void func2(int n, int m){
2     for(int i = 0; i < n; i++){
3         for(int j = 0; j < m; j++){
4             print(i, j);
5         }
6     }
7 }
```

- i Ejecuta  $O(n + m)$  pasos.
- ii Ejecuta  $O(n.m)$  pasos.
- iii Ejecuta  $O(n^2 + m)$  pasos.
- iv Ejecuta  $O(m + n^2)$  pasos.

d (10%) ¿Cuál de las siguientes afirmaciones es correcta con respecto a  $func3(n)$ ?

```

1 void func3(int n){
2     if(n < 1) return;
3     else {
4         System.out.println(n);
5         func3(n - 1);
6     }
7 }
```

- i Esta ejecuta  $T(n) = c + T(n - 1)$  pasos.
- ii Esta ejecuta  $T(n) = n + T(n - 1)$  pasos.
- iii Esta ejecuta  $T(n) = cn + T(n - 1)$  pasos.
- iv Esta ejecuta  $T(n) = c + 2.T(n - 1)$  pasos.

### 3 Notación O 20%

a (10%) Si  $f(m) = O(m^2)$  y  $g(n) = O(n)$  y siempre se cumple que  $m \gg n$  ( $m$  es mucho más grande que  $n$ ), deduzca cuál es el valor de  $O(f(m) + g(n))$ .

- (i)  $O(m^2)$
- (ii)  $O(n + m)$
- (iii)  $O(n^2)$
- (iv)  $O(n \times m)$

b (10%) Si  $f(n) = O(n^2)$  y  $g(n) = O(2^n)$ , deduzca cuál es el valor de  $O(f(n) \times g(n))$ .

- (i)  $O(2^n)$
- (ii)  $O(n^2)$
- (iii)  $O(2^n \times n^2)$
- (iv)  $O(2^n + n^2)$

### 4 Listas 20%

**Nota:** El **add(n)** añade el elemento  $n$  al final de la lista.

**Nota:** El **get(i)** retorna el elemento en la posición  $i$ .

**Nota:** El **size(i)** retorna el tamaño de la lista.

a (10%) ¿Cuál es la complejidad asintótica, en el peor de los casos, del siguiente algoritmo?

```

1 void misterio(int n, LinkedList<
    Integer> lista){
2     for(int i = 0; i < n; i++){
3         for(int j = 0; j < n; j++){
4             lista.add(i + j);
5         }
6     }
7 }
```

- i  $O(n^2)$
- ii  $O(n \log n)$
- iii  $O(n)$
- iv  $O(n^3)$

b (10%) ¿Cuál es la complejidad asintótica, en el peor de los casos, del siguiente método?

```

1 void suma(int n, LinkedList<Integer
    > lista){
2     int i = 0;
3     int suma = 0;
4     while(i < lista.size()){
5         int suma = suma + lista.get(i);
6         i++;
7     }
8 }
```

- i  $O(n^2)$
- ii  $O(n \log n)$
- iii  $O(n)$
- iv  $O(n^3)$