

Estructuras de Datos y Algoritmos 1 - ST0245

Primer Parcial - Jueves (033)

Nombre
Departamento de Informática y Sistemas
Universidad EAFIT

Marzo 19 de 2020

1 Recursión 30%

Una de las tecnologías cruciales para que la Cuarta Revolución Industrial tenga éxito es la *ciberseguridad*. Una de las áreas más importantes de la ciberseguridad es la *criptografía*. Criptografía es el arte y técnica de escribir claves secretas de tal forma que lo escrito solamente sea inteligible para quien sepa descifrarlo.

Consideremos una aplicación de criptografía. Algunas palabras del alfabeto español se pueden escribir como la unión consecutiva de los símbolos de la tabla periódica universal. Por ejemplo, la palabra “Población” se puede escribir como la unión de los símbolos {Po, B, La, C, I, O, N}, PoBLaCION. Te entregan una cadena de caracteres S y un conjunto de símbolos T . El objetivo es determinar si S se puede escribir como la unión consecutiva de cero o más símbolos de T . Por ejemplo, sea $T = \{Ti, B, I, O, C\}$ y $S = \text{“Biótico”}$, la respuesta es verdadero; para $S = \text{“”}$, la respuesta es verdadero; y para $S = \text{“Titan”}$, la respuesta es falso. El siguiente código resuelve el problema, pero le faltan algunas líneas; por favor, complétalas. ¡Gracias!

```
1  boolean solve(String s, List<String> t){
2      return solve(t, s, s.length());
3  }
4  boolean solve(List<String> t, String s,
5      int n){
6      for(int i = 1; i <= n; ++i){
7          String pfx = .....;
8          if(t.contains(pfx)){
9              if(i == n){
10                 return .....;
11             }
12             return .....;
13         }
14     }
15     return false;
16 }
```

- A (10%) Completa la línea 6
- B (10%) Completa la línea 9
- C (10%) Completa la línea 11

Puedes asumir que T contiene todas las posibles combinaciones de acentos, mayúsculas y minúsculas de cada símbolo de la tabla periódica. En Java, el método

`s.substring(a,b)` retorna la subcadena de s entre los índices a y $b - 1$, incluídos. El método `t.contains(s)` retorna verdadero si la cadena s se encuentra dentro de t ; de lo contrario, falso.

2 Notación O 20%

En la vida real, empresas como *Riot Games* hacen un uso extensivo de bases de datos. En muchas bases de datos, los tiempos de búsqueda son $O(\log n)$.

- A (10%) Sea c una constante, $f(n)$ es $O(g(n))$ si $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c$, donde $0 < c < \infty$.
- (a) Verdadero
- (b) Falso
- B (10%) Determina cuales de las siguientes proposiciones son verdaderas. Varias pueden ser verdaderas.
- (a) Si $f(n)$ es $O(g(n))$ y $g(n)$ es $O(h(n))$, entonces $f(n)$ es $O(h(n))$.
- (b) Para cualquier base $b > 0$, $\log_b(n)$ es $O(\ln(n))$.
- (c) Para cualquier constante c , c es $O(c)$.
- (d) Sean $f_1(n) = O(g_1(n))$ y $f_2(n) = O(g_2(n))$. Entonces, $(f_1 + f_2)(n) = O(\max(|g_1(n)|, |g_2(n)|))$

Recuerda la regla de cambio de base de los logaritmos: $\log_a n = \frac{\log_c n}{\log_c a}$ y no olvides que $\ln n = \log_e n$

3 Listas enlazadas 30%

En la vida real, las listas enlazadas se utilizan para implementar el mecanismo de deshacer y rehacer en *Microsoft Word* y *Adobe Photoshop*. Además, de acuerdo a *Interview-Bit*, el siguiente problema es frecuente en la primera entrevista técnica para práctica laboral en *Amazon*. Dado una lista simplemente enlazada **head** y un entero $k \geq 0$, Amazon requiere rotar k posiciones los nodos de **head**. Por ejemplo, si $\text{head} = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow \text{null}$ y $k = 2$, la respuesta es $\text{result} = 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \text{null}$. El siguiente código entrega la solución, pero faltan algunas líneas; por favor, complételas.

```

1  class Node {Node next; int data;}
2  Node solve(Node head, int k){
3      Node slow = head, fast = head;
4      if(head == null) return null;
5      Node counter = head;
6      int elements = 0;
7      while(counter != null){
8          elements ++;
9          counter = counter.next;
10     }
11     k = .....;
12     for(int i = 0; i < k; ++i){
13         fast = fast.next;
14     }
15     while(fast.next != null){
16         slow = slow.next;
17         fast = fast.next;
18     }
19     fast.next = .....;
20     Node result = slow.next;
21     slow.next = null;
22     return result;
23 }

```

- A (10%) Completa la línea 11
- B (10%) Completa la línea 19
- C (10%) ¿Cuál es la complejidad asintótica, en el peor de los casos, del algoritmo anterior?

4 Vectores dinámicos 20%

En la vida real, una variación de un *vector dinámico* (en Java, *ArrayList* y en Python, *list*), llamado *Gap Buffer*, se utilizó para implementar el editor *Emacs*. Dado un *vector dinámico* de enteros a y un valor q , elimina todas las instancias de q en a . Además, devuelve el tamaño del vector dinámico a después de eliminar los elementos. Por ejemplo, si $a = [3, 4, 2, 2, 1, 2, 3, 4, 2]$ y $q = 2$, entonces la respuesta es $a = [3, 4, 1, 3, 4]$ y su tamaño es 5; por consiguiente, `solve` devuelve 5. El siguiente código ejecuta la operación, pero faltan algunas líneas; por favor, complételas.

```

1  int solve(ArrayList<Integer> a, int q){
2      int next = 0;
3      for(int i = 0; i < a.size(); ++i){
4          if(a.get(i) == q){
5              next = .....;
6          }else{
7              a.set(i - next, a.get(i));
8          }
9      }
10     for(int i = 0; i < next; ++i){
11         a.remove(.....);
12     }
13     return a.size();
14 }

```

- A (10%) Completa la línea 5
- Y Completa la línea 11

En Java, el arreglo dinámico a se pasa por referencia; siendo así, la idea es no devolver otro arreglo sino modificar el arreglo a y retornar el tamaño del arreglo modificado. El método `a.get(i)` retorna el elemento en la posición i de a ; `a.set(i,x)` asigna el valor x al elemento en la posición i de a y `remove(i)` elimina el elemento en la posición i de a .

5 Complejidad 10%

En la vida real, las grandes empresas de tecnología, solicitan calcular la complejidad de los algoritmos que preguntan en sus entrevistas técnicas; por ejemplo, *Google*, *Facebook* y *Riot Games*. Estas entrevistas se realizan para acceder a una práctica laboral como lo hizo Santiago Zubietta, en 2015, y Juan M. Ciro, en 2019, para hacer sus prácticas en Google y Facebook, respectivamente. ¡Ambos eran estudiantes de la Universidad Eafit! Para prepararnos para esas entrevistas, para cada uno de los siguientes códigos, determina la ecuación que representa su complejidad para el peor caso.

Para el siguiente código, determina la ecuación que representa su complejidad para el peor caso. Considera que C es una constante.

```

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

```

- A $T_1(n, m) = C \times n \times m$
- B $T_2(n, m) = C \times n \times m^2$
- C $T_3(n, m) = C \times n \times m \log m$
- D $T_4(n, m) = C \times n \times m^3$

6 Complejidad (2% extra)

(2 %) En la vida real, los memes se utilizan en presentaciones orales como una estrategia para *romper el hielo*. Con base en los errores que has cometido durante el semestre, escribe un texto para el siguiente meme:

.....

.....

.....

.....

.....

.....

.....



Como un ejemplo, ten en cuenta la regla de la suma, regla del producto, notación O o ecuaciones de recurrencia.