Recuperación del Primer Parcial de PRG - ETSInf
Fecha: 8 de Junio de 2011. Duración: 1 hora

NOTA: Se debe responder en hojas aparte. No es necesario entregar esta hoja.

1. (5 ptos) El siguiente método permite actualizar una `Concordancia` (una lista enlazada de nodos `NodoCnc`) mediante la ocurrencia de una palabra dada `pal` en una línea determinada `numLin`:

```
public void insertar(String pal, int numLin) {
  NodoCnc aux, ant, nuevo;
  boolean found = false;

  aux = primero; ant = null;

  while ((aux != null) && (!found)) {
    if (aux.pal.equals(pal)) {
      aux.numLins.encolar(numLin);
      found = true;
    }
    ant = aux;
    aux = aux.siguiente;
  }
  if (!found) {
    nuevo = new NodoCnc(pal, numLin);
    if (primero == null) primero = nuevo;
    else ant.siguiente = nuevo;
    talla++;
  }
 }
}
```

Asumiendo que el acceso a los atributos (de `Concordancia` y de `NodoCnc`) se realiza en tiempo constante, y que el coste asociado tanto al constructor de objetos `NodoCnc` como a las operaciones `equals` y `encolar` (clases `String` y `ColaIntEnla`, respectivamente) es también constante, se pide:

a) Indicar cuál es el tamaño o talla del problema, así como la expresión que lo representa.

b) Identificar, caso de que las hubiere, las instancias del problema que representan el caso mejor y peor del algoritmo.

c) Elegir una unidad de medida para la estimación del coste (pasos de programa, instrucción crítica) y acorde con ella, obtener una expresión matemática, lo más precisa posible, del coste temporal del programa, a nivel global o en las instancias más significativas si las hay.

d) Expresar el resultado anterior utilizando notación asintótica.

Solución:

a) El tamaño o talla del problema es el número de elementos de la concordancia, es decir, el número de palabras de la lista. Se expresa mediante el atributo `talla`, denotado en adelante como n.

b) Para una misma talla sí que presenta instancias distintas. El *caso mejor* se da cuando la palabra a insertar ya se encuentra en la concordancia y además es la primera de la lista. El *caso peor* ocurre cuando la palabra no se encuentra en la concordancia.

c) Dado que todas las instrucciones del algoritmo tienen coste constante, se escoge la comparación `aux.pal.equals(pal)` como *instrucción crítica* de éste, al estar situada en el cuerpo del bucle. En el *caso mejor*, el método termina en la primera iteración del bucle, dado que la búsqueda concluye con éxito en el primer nodo de la lista, realizando una única comparación, es decir, $T^m(n) = 1$. En el *caso peor*, el bucle se ejecuta un total de n iteraciones (una por cada uno de los nodos de la lista), y por lo tanto, $T^p(n) = \sum_{k=0}^{n-1} 1 = n$.

d) En notación asintótica: $T^m(n) \in \Theta(1)$ y $T^p(n) \in \Theta(n)$. Por tanto, $T(n) \in \Omega(1)$ y $T(n) \in O(n)$, es decir, el coste temporal está acotado inferiormente por una función constante y superiormente por una función lineal con el número de palabras de la concordancia.

2. (5 ptos) Dado el siguiente método **recursivo** cuya llamada inicial se debe realizar con: $palindromo(s, 0, s.length() - 1)$:

```
public static boolean palindromo( String s, int ini, int fin )
{
  if (ini>=fin) return true;
  if (s.charAt(ini)!=s.chartAt(fin)) return false;
  return palindromo(s,ini+1, fin-1);
}
```

Se pide:

a) Indicar cuál es el tamaño o talla del problema, así como la expresión que lo representa.

b) Identificar, caso de que las hubiere, las instancias del problema que representan el caso mejor y peor del algoritmo.

c) Elegir una unidad de medida para la estimación del coste (pasos de programa, instrucción crítica) y acorde con ella, obtener una expresión matemática, lo más precisa posible, del coste temporal del programa, a nivel global o en las instancias más significativas si las hay.

d) Expresar el resultado anterior utilizando notación asintótica.

Solución:

a) El tamaño o talla del problema es el número de carácteres de la cadena s y la expresión que lo representa es $fin - ini + 1 = s.length()$. En adelante, denominaremos a ese número n. Esto es, $n = fin - ini + 1 = s.length()$.

b) Para una misma talla sí que presenta instancias distintas. El *caso mejor* se da cuando el primer carácter de la cadena es distinto al último carácter de la cadena. El *caso peor* ocurre cuando la cadena es un palíndromo (capicua para las letras).

c) Resolvemos el coste por recurrencia: En el *caso mejor* sólo se ejecutará una vez y la función de coste temporal en este caso será $T^m(n) = 1$. En el *caso peor* tendremos la siguiente función $T^p(n) = T^p(n-2) + k$ si $n > 1$ y $T^p(n) = k'$ si $n = 0$ o $n = 1$ que resolviendo por sustitución obtenemos un coste de $T^p(n) = \frac{n}{2}k + k'$.

d) En notación asintótica: $T^m(n) \in \Theta(1)$ y $T^p(n) \in \Theta(n)$. Por tanto, $T(n) \in \Omega(1)$ y $T(n) \in O(n)$, es decir, el coste temporal está acotado inferiormente por una función constante y superiormente por una función lineal con la talla del problema.