

Recuperació del Primer Parcial de PRG - ETSInf

Data: 8 de Juny de 2011. Duració: 1 hora

NOTA: Heu de respondre en fulls apart. No cal entregar aquest full.

1. (5 punts) El següent mètode permet actualitzar una **Concordancia** (una llista de nodes (u objectes) **NodeCnc**) mitjançant una ocurrència d'una paraula donada **par** en una determinada línia **numLin**:

```
public void inserir( String par, int numLin ) {
    NodeCnc aux, ant, nou;
    boolean trobat = false;

    aux = primer; ant = null;

    while ((aux != null) && (!trobat)) {
        if (aux.par.equals(par)) {
            aux.numLins.encuar(numLin);
            trobat = true;
        }
        ant = aux;
        aux = aux.seguint;
    }
    if (!trobat) {
        nou = new NodeCnc( par, numLin );
        if (primer == null) primero = nou;
        else ant.seguint = nou;
        talla++;
    }
}
```

Assumint que el cost d'accedir als atributs dels objectes de les classes **Concordancia** i **NodeCnc** és constant, que el cost de crear un nou objecte de la classe **NodeCnc** amb qualsevol dels constructors de la classe és constant, i que el cost de les operacions **equals()** de la classe **String** i **encuar()** de la classe **CuanIntEnlla** també es constant, us demanem:

- Indicar quina és la grandària o talla del problema, així com l'expressió que la representa.
- Identificar, en cas que n'hi haja, les instàncies del problema que representen els casos millor i pitjor de l'algorisme.
- Triar una unitat de mesura per estimar el cost (pas de programa o instrucció crítica), i d'acord amb la mesura escollida obtenir l'expressió matemàtica, el més precisa possible, del cost temporal del programa. A nivell general si no existeixen instàncies significatives, o per als casos millor i pitjor en cas de que sí en existeixen.
- Expressar el resultat anterior fent ús de la notació asimptòtica.

Solució:

- La grandària o talla del problema és el nombre d'elements de la concordança, és a dir, el nombre de paraules de la llista. S'expressa mitjançant l'atribut **talla**, denotat d'ací endavant com *n*.
- Per una mateixa talla sí que presenta instàncies distintes. El *cas millor* es dona quan la paraula a insertar ja es troba a la concordança i a més és la primera de la llista. El *cas pitjor* es dona quando la paraula no es troba a la concordança.

- c) Donat que totes les instruccions de l'algorisme tenen cost constant, es tria la comparació `aux.pal.equals(pal)` com *instrucció crítica*, per estar situada al cos del bucle. En el *cas millor*, el mètode acaba en la primera iteració del bucle, donat que la cerca conclou amb èxit en el primer node de la llista, realitzant una única comparació, és a dir, $T^m(n) = 1$. En el *cas pitjor*, el bucle s'executa un total de n iteracions (una per cadascun dels nodes de la llista), i, per tant, $T^p(n) = \sum_{k=0}^{n-1} 1 = n$.
- d) En notació asimptòtica: $T^m(n) \in \Theta(1)$ i $T^p(n) \in \Theta(n)$. Per tant, $T(n) \in \Omega(1)$ i $T(n) \in O(n)$, és a dir, el cost temporal està acotat inferiorment per una funció constant i superiorment per una funció lineal amb el nombre de paraules de la concordança.

2. (5 punts) Donat el següent mètode **recursiu** on la seua invocació inicial ha de ser: `palindrom(s, 0, s.length()-1)`:

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

Es demana:

- Indicar quina és la grandària o talla del problema, així com l'expressió que la representa.
- Identificar, en cas que n'hi haja, les instàncies del problema que representen els casos millor i pitjor de l'algorisme.
- Triar una unitat de mesura per estimar el cost (pas de programa o instrucció crítica), i d'acord amb la mesura escollida obtenir l'expressió matemàtica, el més precisa possible, del cost temporal del programa. A nivell general si no existeixen instàncies significatives, o per als casos millor i pitjor en cas de que sí en existeixen.
- Expressar el resultat anterior fent ús de la notació asimptòtica.

Solució:

- La grandària o talla del problema és el nombre de caràcters de la cadena s i l'expressió que la representa és $fin - ini + 1 = s.length()$. D'ací endavant, anomenarem a aquest número n . Açò és, $n = fin - ini + 1 = s.length()$.
- Per una mateixa talla sí que presenta instàncies distintes. El *cas millor* es dona quando el primer caràcter de la cadena és distint de l'últim caràcter de la cadena. El *cas pitjor* es dona quan la cadena és un palíndrom (cap-i-cua per a les lletres).
- Resolent el cost per recurrència: En el *cas millor* només s'executarà una vegada i la funció de cost temporal en aquest cas serà $T^m(n) = 1$. En el *cas pitjor* tindrem la següent funció $T^p(n) = T^p(n-2) + k$ si $n > 1$ i $T^p(n) = k'$ si $n = 0$ o $n = 1$ que resolent per substitució obtenim un cost de $T^p(n) = \frac{n}{2}k + k'$.
- En notació asimptòtica: $T^m(n) \in \Theta(1)$ i $T^p(n) \in \Theta(n)$. Per tant, $T(n) \in \Omega(1)$ i $T(n) \in O(n)$, és a dir, el cost temporal està acotat inferiorment per una funció constant i superiorment per una funció lineal amb la talla del problema.