UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

etsinf

Tema 2. Análisis

Programación (PRG)

Jorge González Mollá

Departamento de Sistemas Informáticos y Computación

Índice

Instancias

- El coste de un algoritmo es una función no decreciente T(n) parametrizada sobre la talla del problema n.

- Además del factor cuantitativo de la talla, dicho coste también puede depender de la existencia o no de diferentes instancias, es decir, de que los datos de entrada tengan una forma u otra.

- Una instancia de un problema representa todas aquellas configuraciones distintas de los datos de entrada de una misma talla, para las cuales el coste del algoritmo es similar, es decir, la lógica del programa se comporta del mismo modo.

Caso Mejor, Peor, y Promedio

- En el análisis de un algoritmo, si se detectan diferentes instancias que causan variaciones significativas en su comportamiento, entonces se tipifican los siguientes casos:
  - Coste del algoritmo en el Caso Mejor: es la complejidad del mismo para aquella instancia del problema con coste mínimo. Se denota por $T^m(n)$.
  - Coste del algoritmo en el Caso Peor: es la complejidad del mismo para aquella instancia del problema con coste máximo. Se denota por $T^p(n)$.
  - Coste del algoritmo en el Caso Promedio: es la media de los costes de todas las instancias del problema. Se denota por $T^\mu(n)$.
- A la hora de analizar un algoritmo en general, los costes que más interesan son sus comportamientos en el peor caso y en promedio.

Notación asintótica

Procedimiento para calcular el coste teórico o a priori de un algoritmo:

1. Determinar la talla del problema, es decir, estudiar qué parámetro(s) de entrada, de tipo cuantitativo, influyen en el coste.
2. Analizar si existen instancias significativas en cuanto al coste, independientes de la talla del problema. Es decir, se observan comportamientos distintos del algoritmo, que son ajenos a la talla.
3. Obtener la función de coste:
   - Si no hay instancias significativas, se utiliza la notación Θ.
   - Si existen instancias significativas, se analizan por separado los casos mejor y peor. El análisis del caso mejor permite establecer la cota inferior del coste del algoritmo, mientras que el del caso peor permite establecer la cota superior del coste del algoritmo.

     Para la cota inferior (caso mejor) se utiliza la notación Ω.

     Para la cota superior (caso peor) se utiliza la notación O.

Recorrido

- Sea a un array de n elementos que se puede recorrer mediante un algoritmo como el siguiente:

```
for (int i = 0; i < n; i++)
    tratar(a[i]);
```

- [Asunción]: `tratar(a[i]);` es una operación de coste constante.

- La talla del problema es el número de elementos del array n.

- El problema de recorrido de un array sólo presenta una instancia, por tanto, no se puede distinguir ni un caso mejor ni un caso peor. El coste genérico del algoritmo se aproxima contando el número de veces que se ejecuta el cuerpo del bucle (`tratar(a[i]);` y `i++`).

- El coste del algoritmo es: $T(n) = \sum_{i=0}^{n-1} 1 = n$ $T(n) \in \Theta(n)$

Búsqueda Lineal

- Sea a un array de n elementos sobre el que se realiza la búsqueda del primer elemento que cumpla una determinada propiedad:

```
int i = 0;
while ((i < n) && !(PROPIEDAD(a[i]))) { i++; }
if (i < n) return i;
else       return -1;
```

- [Asunción]: `PROPIEDAD(a[i])` es la propiedad booleana que debe cumplir el elemento `a[i]` que buscamos; tiene un coste constante.

- La talla del problema es el número de elementos del array n.

Búsqueda Lineal

- El problema de búsqueda en un array sí presenta varias instancias:
  - el elemento a buscar se encuentra en la posición 0
  - el elemento a buscar se encuentra en la posición 1
  - . . .
  - el elemento a buscar se encuentra en la posición n-1
  - el elemento a buscar no se encuentra

Con éxito

Sin éxito

A priori hay n+1 instancias

- Como el algoritmo realiza la búsqueda secuencial de forma ascendente (de izquierda a derecha, y desde la posición 0 del array):
  - Caso Mejor: el elemento a buscar está en la posición 0 del array.
  - Caso Peor: la búsqueda fracasa (o tiene éxito en la posición n-1).

Búsqueda Lineal

- El coste del algoritmo depende del número de elementos del array.
- Pero también depende de la instancia del problema.
- Para este algoritmo, en total, hay n instancias significativas.
- Considerando a `!(PROPIEDAD(a[i]))` y a `i++;` cuerpo del bucle.

  - Caso Mejor: $T^m(n) = 1$
  - Caso Peor: $T^p(n) = \sum_{i=0}^{n-1} 1 = n$

  - Cota inferior: $T(n) \in \Omega(1)$ CONSTANTE
  - Cota superior: $T(n) \in O(n)$ LINEAL