Diseño y Análisis de Algoritmos
Introducción

Dr. Jesús Ariel Carrasco Ochoa
ariel@inaoep.mx
Oficina 8311

Contenido

- Algoritmo
- Análisis de Algoritmos
- Diseño de Algoritmos

Algoritmo

Una secuencia de pasos para transformar una entrada en una salida.

Algoritmo

Procedimiento que toma un conjunto de valores como entrada y produce un conjunto de valores como salida.

- ¿ Cuál es la diferencia con una función en matemáticas ?

Algoritmo

- Serie de pasos para resolver un problema.
  - **Finito**
    - En pasos
    - En proceso
  - **Bien definido**
    - En todo momento se debe saber qué hacer

Pseudocódigo

- Usaremos una notación similar a C (if, while, for)
- Usaremos *Repeat-Until* como en pascal

Ejemplo: Ordenamiento de Burbuja

```
Entrada: a un arreglo de números
         N el tamaño del arreglo

Salida:  a ordenado
         Para todo i=1,…,N-1 a[i]<=a[i+1]
```

```
for( i=1 ; i<N ; i++)
      for( j=0 ; j<(N-i) ; j++)
            if( a[j] > a[j+1] )
              {
                temporal = a[j+1];
                a[j+1] = a[j];
                a[j] = temporal;
              }
```

Ejemplo: Quicksort

```
Entrada: a un arreglo de números
         ini, fin índices inicial y final de a

Salida:  a ordenado
         Para todo i=1,…,N-1 a[i]<=a[i+1]

izq = ini;
der = fin;
pivote = a[(izq+der)/2];
```

Ejemplo: Quicksort

```
Repeat
      while( a[izq]<pivote && izq<fin )izq++;
      while( pivote<a[der] && der>ini )der--;
      if(izq <= der)
          {
            temporal=  a[izq];
            a[izq]= a[der];
            a[der]=temporal;
            izq++;
            der--;
          }
until (izq > der);
```

Ejemplo: Quicksort

```
if( ini < der )
  qs( a, ini ,der );

if( fin > izq )
  qs( a, izq, fin);
```

Análisis de Algoritmos

Estimar los recursos que un algoritmo requiere para ejecutarse

- Tiempo
  - número de operaciones (sumas, intercambios, etc.)
- Espacio
  - número de unidades de almacenamiento (bytes, palabras, etc.)

Análisis de Algoritmos

La estimación se hace relativa al tamaño del problema

La forma de medir el tamaño de un problema depende del problema específico

- Ordenar
  - Número elementos a ordenar n
- Multiplicación de matrices
  - Dimensiones de las matrices a multiplicar nxk y kxm

Análisis de Algoritmos

En algunos casos no es obvio

- Calcular el n-esimo número primo
  - ?
- Calcular la forma normal disyuntiva reducida de una función Booleana
  - ?
- Obtener las raíces de un polinomio de grado 2
  - ?

Análisis de Algoritmos

En algunos casos no es posible determinar exactamente el tiempo y espacio que requiere un algoritmo.

Algunas veces esto depende no solamente del tamaño del problema sino de la forma del mismo

- Calcular todos los reductos de un sistema de decisión
  - En algunos casos para sistemas de decisión del mismo tamaño encontrar todos los reductos puede tomar tiempo y espacio muy diferentes usando un mismo algoritmo

Análisis de Algoritmos

En estos casos se analizan por separado:

- Peor caso
  - Es una cota superior, nunca tardará más o requerirá más espacio
- Mejor caso
  - Es una cota inferior, nunca tardará menos o requerirá menos espacio
- Caso promedio
  - Es el tiempo/espacio esperado

Análisis de Algoritmos

El caso promedio no es:

(Mejor-caso + Peor-caso) / 2

El caso promedio tiene que ver con el tiempo/espacio de cada caso posible y la probabilidad de que se presente cada caso.

- Usualmente calcular el caso promedio es muy difícil
- El mejor caso no es útil
- Comúnmente se calcula el peor caso

Diseño de Algoritmos

Dado un problema, crear un algoritmo para resolverlo

- Existen diferentes técnicas
  - Algoritmos ávidos
  - Divide y vencerás
  - Programación dinámica
  - etc.

Diseño y Análisis de Algoritmos
Introducción

Dr. Jesús Ariel Carrasco Ochoa
ariel@inaoep.mx
Oficina 8311