Dividir y conquistar

Pedro O. Pérez M., MTI

Análisis y diseño de algoritmos
Tecnológico de Monterrey

pperezm@tec.mx

02-2019

Contenido

Definición

Es una técnica que permite encontrar la solución de un problema descomponiéndolo en subproblemas más pequeños (dividir) y que tienen la misma naturaleza del problema original, es decir, son similares a este. Luego resuelve cada uno de los subproblemas recursivamente hasta llegar a problemas de solución trivial o conocida con antelación (conquistar) para, finalmente, unir las diferentes soluciones (combinar) y así conformar la solución global al problema.

Forma general

Procedure 1 DIVIDE_AND_CONQUER

Input: X
  if X is simple or known **then**
    return $SOLUTION(X)$
  else
    Decompose X into smaller problems $x_1, x_2, ..., x_n$
    for $i \leftarrow 1$ **to** n **do**
      $y_i \leftarrow DIVIDE\_AND\_CONQUER(X_i)$
    end for
    Combine the y_i to get the Y that is solution of X
    return Y
  end if

Búsqueda binaria

Buscar un elemento x en un arreglo ordenado A de n elementos.

Procedure 2 BINARY_SEARCH

Input: $A : Array, low : Index, high : Index, k : Key$

```
if low > high then
    return −1
else
    mid ← FLOOR((high + low)/2)
    if k = A[mid] then
        return mid
    else if k < A[mid] then
        return BINARY_SEARCH(A, low, mid − 1, key)
    else if k > A[mid] then
        return BINARY_SEARCH(A, mid + 1, high, key)
    end if
end if
```

Permutaciones

Hallar todas las permutaciones de un número. Por ejemplo, las permutaciones de 123 son: {123, 231, 321, 312, 132, 213, 123}.

Procedure 3 PERMUTATION

Input: $S : String, pos : Integer$

```
if pos > 1 then
  for i ← 1 to pos do
    SWAP(number, i, pos)
    PERMUTATION(number, pos − 1)
    SWAP(number, i, pos)
  end for
else
  print S
end if
```

Torres de Hanoi

En este problema hay tres ejes verticales. En uno de los ejes se acomoda un número indeterminado de discos, todos de diferente tamaño y ordenados de abajo hacia arreglo del más grande al más pequeño. El reto consiste en mover todos los discos del eje en el que se encuentran a un eje destino utilizando el otro eje como auxiliar, de acuerdo con las siguientes reglas:

- Solo se puede mover un disco a la vez.
- Solo se pueden mover los discos que están en los topes de los ejes.
- No puede quedar un disco más grande sobre uno más pequeño.

El problema de pasar *n* discos del eje inicial al eje final se puede dividir en el problema de pasar *n - 1* discos del inicial a un eje auxiliar, luego pasar un disco al poste final y finalmente pasar los *n - 1* del eje auxiliar al final (con el mismo algoritmo).

Procedure 4 HANOI

Input: $n : Integer, start : Index, aux : Index, end : Index$

```
if n > 0 then
    HANOI(n − 1, start, end, aux)
    print Move from start to end
    HANOI(n − 1, aux, start, end)
end if
```

Exponenciación rápida

Dados dos números, x y n, calcular el resultado de x^n, haciendo uso de la técnica de dividir y conquistar.

Procedure 5 FAST_POW

Input: $x : Real, n : Integer$
  if $n < 0$ **then**
    return $FAST\_POW(1/x, -n)$
  else if $n == 0$ **then**
    return 1
  else if $n == 1$ **then**
    return x
  else if $n \mod 2 = 0$ **then**
    return $FAST\_POW(x * x, n/2)$
  else if $n \mod 2 = 1$ **then**
    return $x * FAST\_POW(x * x, (n-1)/2)$
  end if

Prefijo común más largo

Dado un conjunto de cadenas, encontrar el prefijo común más largo. Por ejemplo, dadas la siguiente cadenas: “geeksforgeeks”, “geeks”, “geek”, “geezer”, el resultado esperado es: "gee"[1].

[1] https://goo.gl/rqjV76

Procedure 6 FIND_PREFIX

Input: $A : String, B : String$

```
result ← ""
i ← 1
j ← 1
while i < A.length and j < B.length do
    if A[i] <> B[j] then
        break
    end if
    result ← result + A[i]
    i ← i + 1
    j ← j + 1
end while
return result
```

Procedure 7 COMMON_PREFIX

Input: $A : Array, low : Index, high : Index$
  if $low == high$ **then**
    return $A[low]$
  end if
  if $low < high$ **then**
    $mid \leftarrow (high + low)/2$
    $str1 \leftarrow COMMON\_PREFIX(A, low, mid)$
    $str2 \leftarrow COMMON\_PREFIX(A, mid + 1, high)$
    return $FIND\_PREFIX(str1, str2)$
  end if