

Dividir y conquistar

Pedro O. Pérez M., MTI

Análisis y diseño de algoritmos
Tecnológico de Monterrey

pperezm@tec.mx

02-2019

Contenido

Introducción

Ejemplos clásicos

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, \dots, 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

$mid \leftarrow FLOOR((high + low)/2)$

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  $i \leftarrow 1$  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
