

Problema del Agua en marte  
CIT 2001 Diseño y Análisis de Algoritmos  
Pablo Veliz  
16 de septiembre de 2016

El problema consiste en encontrar un numero que sean menor o igual a su antecesor y sucesor (mínimo local), se dispone de un arreglo de enteros con orden desconocido y se solicita encontrar una solución en tiempo  $O(\log(n))$ .

La mejor forma para abordar este ejercicio es acceder a la mitad del arreglo y luego revisar el número anterior y el sucesor de este, si el antecesor es menor, se realizara la misma búsqueda pero en el sub-arreglo [inicio, mitad], en caso de que el sucesor sea el menor, se buscará en el arreglo [mitad, final] y así sucesivamente hasta encontrar un número que cumpla la condición. En caso de que la mitad sea igual al inicio o al final, no habrá encontrado un mínimo local.

```
1 def findlocalR(A, mid, low, hi):
2     if mid >= hi or low >= hi:
3         return -1
4     if A[mid-1] >= A[mid] and A[mid+1] >= A[mid]:
5         return A[mid]
6     elif A[mid-1] > A[mid] and A[mid+1] < A[mid]: # \
7         return findlocalR(A, mid/2, mid, hi)
8     else:
9         return findlocalR(A, mid/2, 0, mid)
10
11 def findlocal(A):
12     return findlocalR(A, len(A)/2, 0, len(A))
13
```

La forma de demostrar este algoritmo, probaremos con el método de inducción que para  $T(n)$ . Se ejecutara el algoritmo generando el mismo pero con la mitad de el largo y se usara una comparación constante, de esta forma:

$$T(n) = T(n/2) + O(1) \quad O(1) = C$$

Esto se cumple para la primera iteración, ahora, para  $n/2$ :

$$T(n/2) = T(n/4) + O(1)$$

$$T(n) = (T(n/4) + C) + C$$

Dando como resultado:

$$T(n) = T(n/4) + C$$

Ahora, para  $N/N$  (caso general) se tendrá la ecuación:

$$T(n/n) = T(n/n) + C + C + C + \dots$$

$C$  se ejecutara una cantidad  $\log N$ , por lo tanto:

$$T(n/n) = c \log |N|$$

Esto comprueba que para cualquier  $n$ , el tiempo de ejecución será  $T(n) = \log |N|$ .

Este método comparará los números (costo  $O(c)$ ) y luego se ejecutara nuevamente con un sub-arreglo de tamaño  $N/2$  una cantidad  $O(\log |N|)$  de veces. Dada la ecuación:

$$T(n) = O(c) + O(\log |N|)$$

lo que da como resultado final:

$$T(n) = O(\log |N|)^1$$