

Correction TD3

Algorithm I



Matthieu Jimenez

Hiver 2015

Correction TD3

Algorithm I

Exercice I

Proposer un algorithme de recherche dichotomique:

Correction:

```
int search(left, right : int ; x : int){
    // teste si x se trouve dans A[left], A[left+1], .. A[right]
    // retourne la position de x dans A, ou 0 si x ne se trouve pas dans A
    if (left>right) then
        return 0 ; //exit - x pas trouvé
    m = (left+right)/2 ; // integer division - resultat est un entier
    if (A[m]==x) then return m ; // exit - on l'a trouvé
    else if (x< A[m]) then
        return search(left,m-1,x) ;
            else \\{x> A[m]}
    return search(m+1,right,x) ;
}
```

Exercice II

Enoncé:

Résoudre les relations de récurrence suivantes:

1. $T(n) = T(n-1) + 3, T(1) = 1$

2. $T(n) = T(n/2) + 1, T(1) = 1$

Correction:

(a) $T(n) = T(n-1) + 3, T(1) = 1.$

$$T(n) = T(n-1) + 3 = T(n-2) + 6 = \dots = T(n-(n-1)) + 3(n-1) = T(1) + 3(n-1) = 1 + 3(n-1) = 3n - 2 = O(n)$$

(b) $T(n) = T(n/2) + 1, T(1) = 1$ (supposer $n = 2^k$)

Soit $n = 2^k$. Alors

$$T(n) = T(2^k) = T(2^{k-1}) + 1 = T(2^{k-2}) + 2 = T(2^{k-k}) + k = T(1) + \log(n) = 1 + \log(n) = O(\log n)$$