

Actividad 3.5

Complejidad de algoritmos recursivos

Algoritmos y Estructuras de Datos

Tema 3: algoritmos recursivos

Calcular el $O(\dots)$ temporal y espacial del peor caso cuando se emplea un algoritmo de búsqueda binaria. Este algoritmo es un método recursivo para saber si un número entero está o no está en un array de enteros previamente ordenado (ver “Búsqueda dicotómica” en http://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda).

Una posible implementación de dicho algoritmo en C++ es la siguiente:

```
bool busqueda_binaria(const int v[], int principio, int fin, int &x){
    bool res;
    if(principio <= fin){
        int m = (fin + principio)/2;
        if(x < v[m]) res = busqueda_binaria(v, principio, m-1, x);
        else if(x > v[m]) res = busqueda_binaria(v, m+1, fin, x);
        else res = true;
    }else res = false;
    return res;
}
```

Analiza cuidadosamente el algoritmo, define su función de recurrencia y determina su complejidad computacional.