

```

int bin(int v[], int ini,int final){
    if(final >= ini){
        int mid = (final + ini) / 2;

        if(v[mid] == 0 && v[mid+1] == 1){
            return final-mid;
        }
        else if(v[mid] == 1){
            return final-mid+1 + bin(v,ini,mid-1);
        }else{
            return bin(v,mid+1,final);
        }
    }
    return 0;
}

```

O pior caso para este algoritmo é quando o vetor é completo apenas por 0's ou 1's. Por exemplo: {0,0,0} ou {1,1,1}, pois neste caso o código irá sempre olhar um subvetor a direita ou a esquerda por completo.

Portando:

Para $n = 1$.

$\Theta(1)$.

Para $n > 1$

$T(n/2) + \Theta(1) + \Theta(1)$.

E o melhor caso desse algoritmo é quando o vetor tem a mesma quantidade de 0's e 1's, ou seja, quando ele está dividido ao meio, por exemplo: {0,0,1,1}, pois neste caso o código apenas dividirá o vetor ao meio, irá olhar o primeiro elemento do subvetor a direita e o ultimo do subvetor a esquerda.

Portanto:

Para $n = 1$

$\Theta(1)$.

Para $n > 1$

$\Theta(1) + \Theta(1)$.