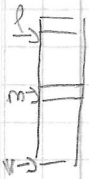
Ex

```
int bsearch (int a[], int N, int x)
{
    int l, u, m;
    l = 0;
    u = N-1;
    while (l <= u) {
        m = (l+u)/2;
        if (a[m] == x) l = u = m;
        else {
            if (a[m] > x)
                then u = m-1;
                else l = m+1;
        }
    }
    if (a[l] == x) return l;
    else return -1;
}
```




variante = u - l

invariante: se estiver no array entre l e u e no fim do ciclo l = u

Pós-condição: retorna -1 se não existe
se existe retorna a posição

Melhor caso: estar no meio $T(N) = \Omega(1)$

Pior caso: Não encontra: $T(N) = 2 + 1 + \sum_{i=1}^{\log_2 N} (5) + 2$

$= 5 + \log_2(N^5) + 1$

$\Theta(\log(N))$

8)

```
int repetidos (int v[], int N)
{
    int rep = 0;
    int i, j;
    for (i=0; (i<N-1) && !rep; i++)
        for (j = i+1; j<N && !rep; j++);
        if (v[i] == v[j]) rep = 1;

    return rep;
}
```

Melhor caso, 1º = 2º $T(N) = 12$

Pior caso: Não há repetidos: $2 + \sum_{i=0}^{n-2} (2 + (\sum_{j=i+1}^{N-1} 3) + 1 + 1) + 1$

$= 4 + \sum_{i=0}^{n-2} 4 + 3(N-i-1) = 4 + 4\sum_{i=0}^{N-2}(N-i) - 3\sum_{i=0}^{N-2} i$