

QUICKSORT - Parte 2: ordenação por Particionamento

$V = [10 \ 8 \ 6 \ 12 \ 13 \ 3 \ 15 \ 4 \ 11 \ 7]$ (QUICKSORT)
 $\theta(n \lg n)$

(1) ↓
6 3 4 7 13 8 15 10 11 12
 (2) ↓ (3) ↓
3 4 6 7 8 10 11 12 15 13
 (4) ↓ (5) ↓ (6) ↓
3 4 6 7 8 10 11 12 13 15
 (5) ↓ (10) ↓

```
void Quicksort(int *V,
               int e, int d) {
    int q;
    if (e < d) {
        q = partition(V, e, d);
        Quicksort(V, e, q-1);
        Quicksort(V, q+1, d);
    }
}
```