Análise de Algoritmos: Quicksort

Aluno: Cristiano Lima Sousa Rosa

```
QUICKSORT(A,p,r)
if p < r then
   q ← PARTITION(A,p,r);
   QUICKSORT(A,p,q-1);
   QUICKSORT(A,q+1,r);
end if
```

```
PARTITION(A,p,r)
x ← A[r];
i ← p-1;
for j ← p to r-1 do
  if A[j] ≤ x then
    i ← i+1;
    trocar(A[i],A[j]);
  end if
end for
trocar(A[i+1],A[r]);
return i+1;
```

```
PARTITION(A,p,r)
x ← A[r];
i ← p-1;
for j ← p to r-1 do
  if A[j] ≤ x then
    i ← i+1;
    trocar(A[i],A[j]);
  end if
end for
trocar(A[i+1],A[r]);
return i+1;
```

```
PARTITION(A,p,r)
x ← A[r];
i ← p-1;
for j ← p to r-1 do
  if A[j] ≤ x then
    i ← i+1;
    trocar(A[i],A[j]);
  end if
end for
trocar(A[i+1],A[r]);
return i+1;
```

$$\sum_{j=p}^{r-1} 1$$

$$r - \cancel{1} - p + \cancel{1}$$

$$= r - p$$

Pior caso

```
QUICKSORT(A,p,r)
if p < r then
    q ← PARTITION(A,p,r);
    QUICKSORT(A,p,q-1);
    QUICKSORT(A,q+1,r);
end if
```

N comparações

N -1 comparações

0 comparações

2	4	8	10

Pivô

2	4	8	10

```
QUICKSORT(A,p,r)
1: if p < r then
2:    q ← PARTITION(A,p,r);
3:    QUICKSORT(A,p,q-1);
4:    QUICKSORT(A,q+1,r);
6: end if
```

1: $\Theta(1)$
2: $\Theta(n)$
3: $\Theta(k)$
4: $\Theta(n-k-1)$

$= T(k)+T(n-k-1)+\Theta(n+1)$

```
QUICKSORT(A,p,r)
1: if p < r then
2:    q ← PARTITION(A,p,r);
3:    QUICKSORT(A,p,q-1);
4:    QUICKSORT(A,q+1,r);
6: end if
```

1: $\Theta(1)$
2: $\Theta(n)$

3: $\Theta(k)$
4: $\Theta(n-k-1)$

$= T(k)+T(n-k-1)+\Theta(n+1)$

Recorrência:
$= T(0)+T(n-0-1)+\Theta(n) = T(n-1)+n$

$$t(n) = t(n-1) + \Theta(n)$$

$$t(n-1) = t(n-1-1) + n-1$$

$$t(n-2) + n-1$$

$$t(n-2) = t(n-2-1) + n-2$$

$$= t(n-3) + n-2$$

$$t(n-3) = t(n-3-1) + n-3$$

$$= t(n-4) + n-3$$

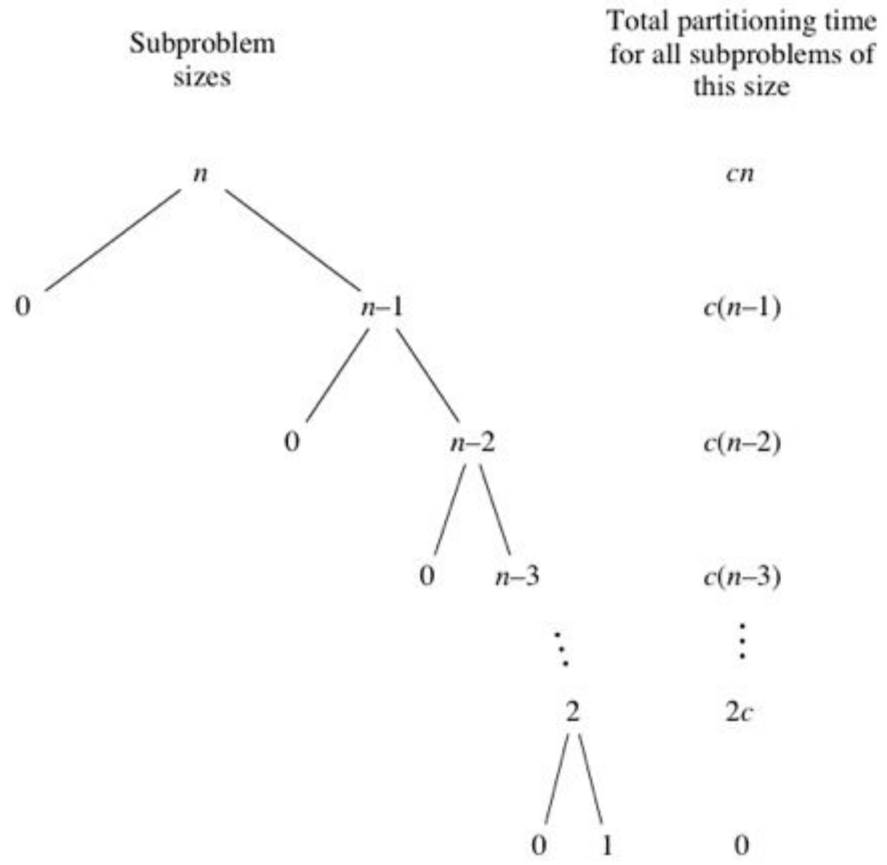
$\dots$

$$t(2) = t(2-1) + 2$$

$$= t(1) + 2$$

$$t(1) = t(0) + 1$$

onde $t(0) = 0$


Subproblem sizes
Total partitioning time for all subproblems of this size
n
cn
0
n–1
c(n–1)
0
n–2
c(n–2)
0
n–3
c(n–3)
2
2c
0
1
0

Sometorio

$$(1+2+3+\dots+n-1)+n$$

$$t(n)=\frac{n(n+1)}{2}=\frac{n^2+n}{2}$$

Melhor caso

```
QUICKSORT(A,p,r)
if p < r then
    q ← PARTITION(A,p,r);
    QUICKSORT(A,p,q-1);
    QUICKSORT(A,q+1,r);
end if
```

N comparações

N/2 comparações

N/2 comparações

2	1	3	5	6

Pivô

```
QUICKSORT(A,p,r)
1: if p < r then
2:    q ← PARTITION(A,p,r);
3:    QUICKSORT(A,p,q-1);
4:    QUICKSORT(A,q+1,r);
6: end if
```

1: $\Theta(1)$
2: $\Theta(n)$

3: $\Theta(n/2)$
4: $\Theta(n/2)$

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

Recorrência:
$= 2T(n/2)+\Theta(n)$

Caso base: $T(1)$

$$t(n) = 2t\left(\frac{n}{2}\right) + n$$

$$t\left(\frac{n}{2}\right) = 2\left[2t\left(\frac{\frac{n}{2}}{2}\right) + \frac{n}{2}\right] + n$$

$$= 2^2 t\left(\frac{n}{2^2}\right) + 2n$$

$$t\left(\frac{n}{2^2}\right) = 2^2\left[2t\left(\frac{\frac{n}{2^2}}{2}\right) + \frac{n}{2^2}\right] + 2n$$

$$= 2^3 t\left(\frac{n}{2^3}\right) + 3n$$

$$t(k) = 2^k t\left(\frac{n}{2^k}\right) + kn$$

$n = ? \rightarrow t\left(\frac{2^k}{2^k}\right) = t(1)$, $n = 2^k$

$2^k + k2^k$

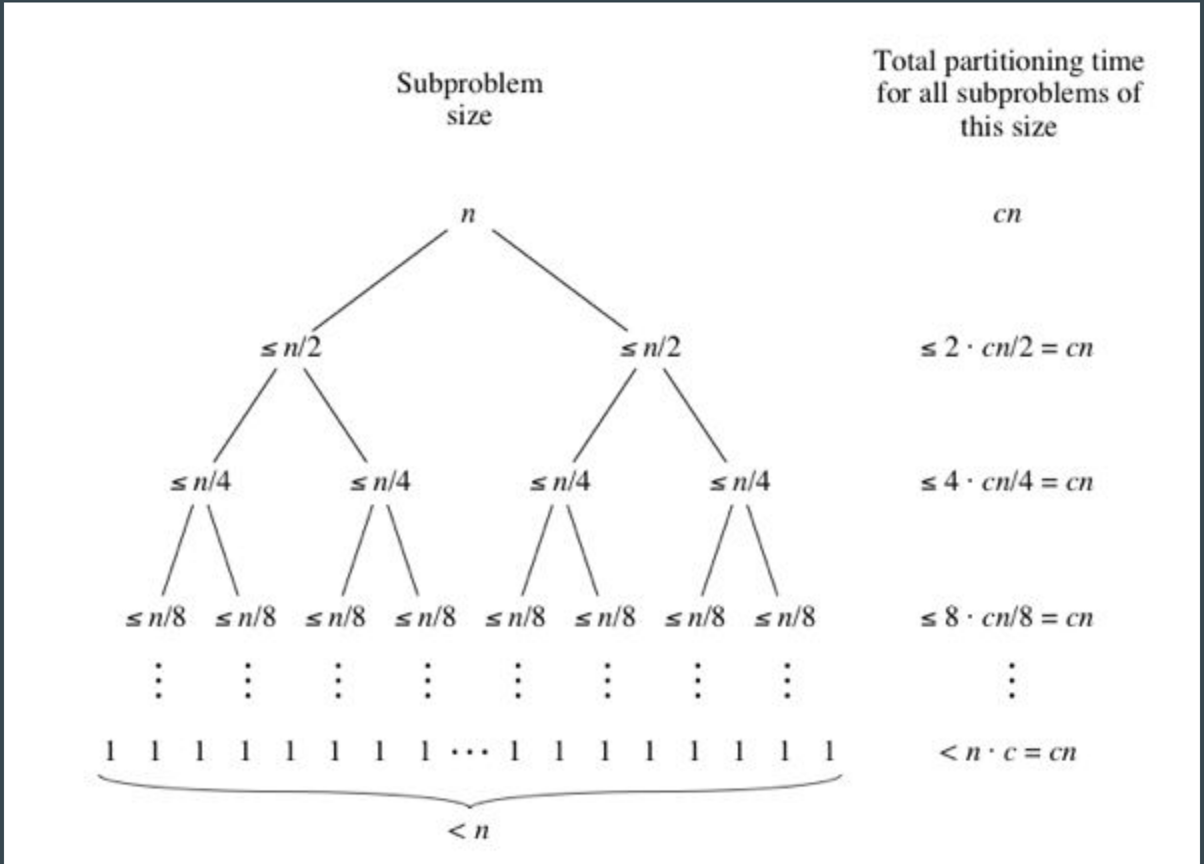
$t(k) = 2^k + k2^k$

$N = 2^k \rightarrow k = \log^n$

$t(n) = 2^{\log^n} + (\log n)2^{\log^n}$

$$\underbrace{t(n) = n + n(\log n)}_{f(n)}$$

Complexidade: $n \log n$


Subproblem size
Total partitioning time for all subproblems of this size
n
cn
≤ n/2
≤ n/2
≤ 2 · cn/2 = cn
≤ n/4
≤ n/4
≤ n/4
≤ n/4
≤ 4 · cn/4 = cn
≤ n/8
≤ n/8
≤ n/8
≤ n/8
≤ n/8
≤ n/8
≤ n/8
≤ n/8
≤ 8 · cn/8 = cn
1 1 1 1 1 1 1 1 ··· 1 1 1 1 1 1 1 1
< n · c = cn
< n

Pior caso	Melhor caso
$O(n^2)$	$n(logn)$

Fim.