Projeto e Análise de Algoritmos

Quicksort

Diane Castonguay
diane@inf.ufg.br

Instituto de Informática
Universidade Federal de Goiás

```
PARTITION(A, p, r)
1.      chave ← A[r]
2.      i ← p – 1
3.      para j ← p até r – 1 faça
4.              se A[j] ≤ chave
5.              então i ← i + 1
6.                      trocar A[i] ↔ A[j]
7.              fim-se
8.      fim-para
9.      trocar A[i+1] ↔ A[r]
10.     retorna i+1
```

```
QUICKSORT(A, p, r)
1. se p < r
2.      então q ← PARTITION(A, p, r)
3.            QUICKSORT(A, p, q – 1)
4.            QUICKSORT(A, q + 1, r)
5.  fim-se
```

```
RANDOMIZED-PARTITION(A, p, r)
1.    i ← RANDOM(p, r)
2.    trocar A[i] ↔ A[r]
3.    retorna PARTITION(A, p, r)

RANDOMIZED-QUICKSORT(A, p, r)
1. se p < r
2.    então q ← RANDOMIZED-PARTITION(A, p, r)
3.          RANDOMIZED-QUICKSORT(A, p, q – 1)
4.          RANDOMIZED-QUICKSORT(A, q + 1, r)
5.  fim-se
```