Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)

Estrutura de Dados I

Professor: Rodrigo Minetto

Exercício 1) Implemente em linguagem C o algoritmo Quick-Sort e o utilize para ordenar 10, 100, 1.000, 10.000, 100.000 e 1.000.000 inteiros em ordem:

- aleatória;
- crescente;
- decrescente;

Para qual dos casos acima o algoritmo foi mais rápido? E para qual caso foi pior? Mostre os tempos do algoritmo, para esses 3 casos. Explique o porquê desses tempos.

Exercício 2) Crie um algoritmo chamado Quick-Find baseado no Quick-Sort para que, em vez de ordenar uma sequência de números inteiros, ele nos retorne o k-ésimo menor elemento dessa sequência. Por exemplo: suponha que os elementos S = {7, 1, 3, 10, 17, 2, 21, 9} estejam armazenados nessa ordem em um vetor e que desejamos obter o quinto maior elemento dessa sequência. Então, uma chamada como Quick-Find (S,0,7,4), deverá retornar o número 9, onde S é o nome do vetor, 0 e 7 são, respectivamente, o menor e o maior índice do vetor e 4 indica que desejamos o quinto menor elemento (0 ... 4). Da mesma forma teste com (S,0,7,2) que deve retornar o valor 3. Obs.: uma solução errada neste exercício é ordenar a sequência e depois tomar o k-ésimo elemento.

Exercício 3) A implementação do algoritmo Quick-Sort que vimos em aula não funciona satisfatoriamente para as seguintes configurações de elementos:

- elementos em ordem crescente;
- elementos em ordem decrescente;

Construa um algoritmo Quick-Sort aleatorizado para melhorar estas execuções. Para tanto sorteie um pivô aleatoriamente dentro de uma partição correta e troque-o com o pivô que sempre é escolhido da última posição. Ou seja:

```
int Particione_Aleatorio (int A[], int l, int r) {
  /*Sorteio aleatório da posição do pivô*/
  -> Sorteio da posição.
  -> Verifique se a posição do sorteio pertence ao intervalo correto.
  -> Faça swap do elemento sorteado com a posição do pivô.
  int x = A[r];
  int i = l - 1;
  ...
}
```

Repita os testes do exercício 1 para verificar as melhorias.