Algoritmos e Estrutura de Dados I - AE22CP - 2013/2

Bruno César Ribas

Exercícios

A. O que é Algoritmo?

QuickSort

A. Escreva o algoritmo da separação

B. Escreva uma função que rearranje um vetor v[p..r] de inteiros de modo que tenhamos v[p..j-1] <= 0 e v[j..r] > 0 para algum j em p..r+1. Faz sentido exigir que j esteja em p..r? Procure fazer uma função rápida que não use vetor auxiliar. Repita o exercício depois de trocar "v[j..r] > 0" por "v[j..r] >= 0". Faz sentido exigir que v[j] seja 0?

C. Um vetor v[p..r] está arrumado se existe j em p..r tal que v[p..j-1] <= v[j] < v[j+1..r] . Escreva um algoritmo que decida se v[p..r] está arrumado. Em caso afirmativo, o seu algoritmo deve devolver o valor de j.

D. Um programador inexperiente afirma que a seguinte implementação da função de separação rearranja o vetor v[p..r], com p < r, e devolve um índice j em p..r-1 tal que v[p..j] <= v[j+1..r].

```
int sep( int v[], int p, int r) {
   int q, i, j, t;
   i = p; q = (p + r) / 2; j = r;
   do {
      while (v[i] < v[q]) ++i;
      while (v[j] > v[q]) --j;
      if (i <= j) {
         t = v[i], v[i] = v[j], v[j] = t;
         ++i, --j;
      }
   } while (i <= j);
   return i;
}
```

Mostre um exemplo onde essa função não dá o resultado esperado. E se trocarmos "return i" por "return i-1"? É possível fazer algumas poucas correções de modo que a função dê o resultado esperado?

E. Qual o resultado da função separa quando os elementos de v[p..r] são todos iguais? E quando v[p..r] é crescente? E quando v[p..r] é decrescente? E quando cada elemento de v[p..r] tem um de dois valores possíveis?

F. A função separa produz um rearranjo estável do vetor, ou seja, preserva a ordem relativa de elementos de mesmo valor?

G. Escreva uma versão recursiva da função separa.

H. Suponha dada uma lista encadeada que armazena números inteiros. Cada célula da lista tem a estrutura abaixo.

```
struct celula {
   int            chave;
   struct celula *prox;
};
```

Escreva uma função que transforme a lista em duas: a primeira contendo as células cuja chave é par e a segunda aquelas cuja chave é ímpar.

I. Implemente o QuickSort

J. Que acontece se trocarmos "p < r" por "p != r" na linha 2 do quicksort (implementado em sala)?

K. Submeta o vetor 77 55 33 99 indexado por 1..4 à função quicksort. Teremos a seguinte sequência de invocações da função (observe a indentação):

```
      quicksort( v,1,4)
          quicksort( v,1,2)
              quicksort( v,1,0)
              quicksort( v,2,2)
          quicksort( v,4,4)
```

Repita o exercício com o vetor 55 44 22 11 66 33 indexado por 1..6.

L. A função quicksort produz uma ordenação estável do vetor?

M. Suponha dada uma implementação da função separa que funciona assim: rearranja v[p..r] e devolve um índice j tal que v[p..j] <= v[j+1..r]. Escreva uma versão do algoritmo Quicksort que use essa implementação do separa. Que restrições devem ser impostas sobre j?

N. Escreva uma versão não recursiva do algoritmo Quicksort. sol

- Exercícios baseados do Professor Paulo Feofiloff

--
Last Modified: Mon Feb 3 16:16:06 2014.