

Introdução à Computação I

CURSO DE C. MOLECULARES – SEGUNDO SEMESTRE DE 2003

Exercício-Programa 9, Peso 1 Data de entrega: até a aula de **99 de novembro de 2003**.

1 Cálculo de centróides

Seja $(w_1; w_2; \dots; w_n)$ uma sequência qualquer de inteiros positivos.

Definição: Um *centróide* para esta sequência é um índice i , tal que $w_1 + w_2 + \dots + w_{i-1}$ é próximo de $w_i + w_{i+1} + \dots + w_n$.

Analogamente, podemos definir centróides para cada uma das subsequências $w_1 + w_2 + \dots + w_{i-1}$ e $w_i + w_{i+1} + \dots + w_n$. Uma decomposição centróide de uma sequência S é a sequência de índices que pode ser usada para decompor S recursivamente até que se tenha sequências de um só elemento.

Por exemplo, dada a sequência:

5, 3, 2, 7, 6, 2, 4, 2, 1, 1

A decomposição centróide é:

(5) (3 7) (2 4 6 8) (9) (10)

que significa que a sequência deve ser primeiro partida na quinta posição, dando:

(5 3 2 7) (6 2 4 2 1 1)

A seguir, na 3a. e 7a. posições, dando:

(5 3) (2 7) (6 2) (4 2 1 1)

e a seguir, na 2a., 4a., 6a. e 8a. posições:

(5) (3) (2) (7) (6) (2) (4) (2 1 1)

e depois:

(5) (3) (2) (7) (6) (2) (4) (2) (1 1)

e depois:

(5) (3) (2) (7) (6) (2) (4) (2) (1) (1)

(Isto pode alternativamente ser visto como: Dado um conjunto de n chaves associadas a frequências (ou pesos) em banco de dados, inserí-las em uma árvore de busca binária de tal modo que o tempo médio de busca é minimizada. Cada centróide nos diz qual parte da sequência deve ir para a sub-árvore esquerda e qual deve ir para a sub-árvore direita.)

Implementar o algoritmo conforme a especificação abaixo que, dada uma sequência de n inteiros positivos em um vetor v , calcula as *posições* entre 1 e n dos centróides e imprime as subsequências à medida que elas vão sendo calculadas. Para o exemplo acima, deve imprimir:

(5 3 2 7) (6 2 4 2 1 1)

(5 3) (2 7) (6 2) (4 2 1 1)
 (5) (3) (2) (7) (6) (2) (4) (2 1 1)
 (5) (3) (2) (7) (6) (2) (4) (2) (1 1)
 (5) (3) (2) (7) (6) (2) (4) (2) (1) (1)

E no final:

(5) (3 7) (2 4 6 8) (9) (10)

Teste com pelo menos **QUATRO** entradas distintas.

Cálculo de centróides:

```

01 centro (vetor v, inteiro inicio, inteiro f im);
02 se inicio + 1 = f im então /* se de nível 0 */
03  imprime(f im) /* por ex. (5 3) */
04 senão
05  meio ← [(f im - inicio)/2];
06  esq ← v[inicio] +      +v[meio - 1];
07  dir ← v[meio] +      +v[f im];
08  achou ← f also;
09  enqto no (achou) faça /* enqto de nível 1 */
10    se esq ≤ dir então /* se de nível 2 */
11      esqD ← esq + v[meio];
12      dirD ← dir - v[meio];
13      se esqD < dirD então
14        /* desloca 1 pos. p/ dir; se de nível 3 */
15        esq ← esqD; dir ← dirD; meio ← meio + 1;
16      senão /* esqD ≥ dirD */
17      se |esq - dir| < |esqD - dirD| então /* se de nível 4 */
18        /* meio é o centróide */
19        imprime(meio); achou ← verdadeiro;
20        se inicio < meio - 1 então centro(v; inicio; meio - 1) fim-se;
21        se meio < f im então centro(v; meio; f im) fim-se;
22      senão
23        /* meio + 1 é o centróide; |esq - dir| ≥ |esqD - dirD| */
24        imprime(meio + 1); achou ← verdadeiro;
25        se inicio < meio então centro(v; inicio; meio) fim-se;
26        se meio + 1 < f im então centro(v; meio + 1; f im) fim-se;
27      fim-se; /* nível 4 */
28    fim-se; /* nível 3 */
29  senão /* esq > dir */
30    esqE ← esq + v[meio - 1];

```

```

28   dirE ← dir - v[meio - 1];
29   se esqE > dirE então
      /* desloca 1 pos. p/ esq.; se de nível 3 */
30   esq ← esqE; dir ← dirE; meio ← meio - 1;
31   senão /* esqE ≤ dirE */
32   se |esqE - dirE| > |esq - dir| então /* se de nível 4 */
      /* meio é o centróide */
33   imprime(meio); achou ← verdadeiro;
34   se inicio < meio - 1 então centro(v; inicio; meio - 1) fim-se;
35   se meio < fim então centro(v; meio; fim) fim-se;
36   senão /* meio - 1 é o centróide */
      /* |esq - dir| ≤ |esqE - dirE| */
37   imprime(meio - 1); achou ← verdadeiro;
38   se inicio < meio - 2 então centro(v; inicio; meio - 2) fim-se;
39   se meio - 1 < fim então centro(v; meio - 1; fim) fim-se;
40   fim-se /* se de nível 4 */
41   fim-se /* se de nível 3 */
42   fim-se /* se de nível 2 */
43   fim-enqto /* enqto 1 */
44   fim-se /* se de nível 0 */

```

Observações

- Este exercício é para ser feito *individualmente*.
- Entregue um envelope com o seu nome e com os seguintes itens:
 - uma descrição simples (cerca de 5 linhas) explicando *como usar* o programa
 - um *disquete* com os seguintes arquivos
 - * o programa em *linguagem C*,
 - * o programa *compilado*,
 - * arquivos com os dados de *entrada* , pelo menos 4 arquivos, chamados ENT1, ENT2, etc., e
 - * arquivos com os dados de *saída*, pelo menos 4 arquivos, correspondentes, chamados SAI1, SAI2, etc.
 - * para *redirecionar* os arquivos para disco, veja o fim da página 9 da apostila.
- Coloque comentários em seu programa explicando o que cada etapa do programa significa! Isso será levado em conta na sua nota.

- **Coloque como comentário o seu nome, número USP, qual o compilador (gcc, TURBO-C, ou outro), qual o sistema operacional (LINUX, MS-DOS, UNIX, ou outro) e qual o modelo de computador (Intel x86, SUN, ou outro) que V usou.**
- Faça uma saída clara! Isso será levado em conta na sua nota.
- Não deixe para a última hora. Planeje investir 70 por cento do tempo total de dedicação em escrever o seu programa todo e simular o programa SEM computador (eliminando erros de lógica) ANTES de digitar e compilar no computador. Isso economiza muito tempo e energia.