CRSC08 Programação Paralela

Atividade 2

“Problema da sequência de soma máxima”

Dada uma sequência de números inteiros A = { a_1, a_2,..., a_n }, determine a subsequência S = { a_i,..., a_j }, com $0 \leq i \leq j \leq n$, que tem o valor máximo para a soma $\sum_{k=i}^{j} a_k$.

Como exemplo considere a sequência: -2, 11, -4, 13, -5, -2.

A solução para essa entrada tem valor 20 com soma no segmento: S = { a_2, ..., a_4 }.

Se obtivermos um valor negativo na resposta, consideramos o valor nulo para simplificar o processamento.

Atividade:

Implemente cada uma das soluções apresentadas em pseudocódigo a seguir. Teste suas implementações com os arquivos 1M.dat, 4M.dat e 8M.dat apresentando o valor máximo encontrado e os limites do seguimento em cada sequencia, conforme exemplo. Em adição, informe o número de operações executadas e o tempo consumido em cada um dos testes (3 soluções aplicadas em 3 arquivos).

Somax2 com complexidade T(n) = O(n^2) … solução que testa cada elemento somado aos subsequentes.

Obs.: precisa adaptar para mostrar os limites da sequencia de soma máxima.

```
Somax2(A, n)
1  max ← 0
2  para i ← 1 até n faça
3       aux  ←  0
4       para  j ← i até n faça
5            aux ← aux + A[j]
6            se aux > max
7               então max ← aux;
8  devolve max
```

Somax3 com complexidade T(n) = O(nlogn)

O algoritmo usa Divisão e Conquista dividindo o problema em dois outros subproblemas iguais(no caso com entrada de n/2). No problema, a subsequência de soma máxima pode ocorrer em um dos seguintes locais:

1. estar inteiramente contida na metade esquerda da entrada,
2. ou inteiramente contida na metade direita da entrada,
3. ou "cruzar" o limite que divide a entrada ao meio.

Os dois primeiros casos podem ser resolvidos recursivamente. O 3º caso pode ser obtido encontrando a maior soma na primeira metade (à esquerda) que inclua o último elemento da 1ªmetade. Em seguida encontra-se a maior soma da segunda metade (à direita) que inclui o primeiro elemento desta. Finalizando, somam-se as duas somas encontradas obtendo "soma máxima que cruza a divisa" do último caso. Como exemplo considere a seguinte entrada:

1ª metade				2ª metade			
4	-3	5	-2	-1	2	6	2

A subsequência de soma máxima para a 1ªmetade é 6 (do elemento a_1 até a_3) e para a 2ªmetade é 8 (do elemento a_6 até a_7). A máxima soma na 1ª metade que inclui o último elemento é igual a 4 (do elemento a_1 até a_4), e na 2ªmetade (que inclui o primeiro elemento da 2ª metade, o a_5) obtemos 7 (do elemento a_5 até a_7). Assim a soma máxima que "cruza" a divisa tem valor: 4 + 7 = 11 (entre a_1 e a_7).

Podemos então, concluir que entre as três somas a última será a resposta para a sequência proposta de oito números: *soma max = 11*.

```
Somax3 (A, e, d)
 1  E_Lim ← D_Lim ← aux_E ← aux_D ← 0
 2  Lim ← (e + d)/2
 3  se e = d
 4       então se A[e] > 0
 5                  então devolve A[e]
 6                  senão devolve 0
 7   Esq  ←  Somax3(A, e, Lim)
 8   Dir  ←  Somax3(A, Lim+1, d)
 9   para i ← Lim até e  faça
10         aux_E ← aux_E + A[i]
11          se aux_E > E_Lim
12             então E_Lim ← aux_E
13   para i ← Lim+1 até d  faça
14          aux_D  ←  aux_D + A[i]
15          se aux_D > D_Lim
16             então D_Lim ← aux_D
17  devolve Máximo(Esq, Dir, E_Lim + D_Lim)
```

Nas chamadas recursivas sempre delimitamos o tamanho da entrada indicando os limites indicando os índices “e”, esquerdo e “d ”, direito, no inicio do processamento de Somax3 temos e = 1 e d = n.

Nas linhas 3 a 6, é manuseado *o caso base*, como na indução matemática. Se $e = d$, há um só elemento, e ele será a soma máxima se não for negativo.

Nas linhas 7 e 8, realizamos duas chamadas recursivas, que dividem o problema inicial em duas partes.

A partir da linha 9 (de 9 a 12 e de 13 a 16) processamos as somas da subseqüência que “cruza o limite” da divisão.

Em seguida com a função **Máximo** selecionamos a maior das somas como listamos anteriormente (à Esquerda, à Direita ou a soma dos máximos “incluindo a divisa”).

Somax4 com complexidade T(n) = O(n) … solução com apenas uma

```
Somax4(A,n)
 1  max ← aux ← j ← k ← 0
 2  ini ← fim ← 1
 3  para i ← 1 até n faça
 4        aux ← aux + A[i]
 5        k ← k + 1
 6        se n = 0
 7              então max ← aux
 8                    fim ← k
 9              senão se aux < 0
10                          então  aux ← 0
11                                 j ← k ← i+1
12                                 ini ← j
13  devolve (max, ini, fim)
```