1

Lista de Exercícios

Edson Norberto Cáceres
Henrique Mongelli
DCT - UFMS,
79070-900 - Campo Grande, MS, Brasil,
edson@dct.ufms.br,
mongelli@dct.ufms.br.

1 Lista de Exercícios

1. Desenvolva um algoritmo que encontre o maior valor de um conjunto S com n números no modelo BSP/CGM com p processadores. Analise a complexidade do algoritmo proposto. Modifique o algoritmo para o problema de encontrar os dois maiores valores. Generalize para o caso de computar os k maiores elementos do conjunto. Estime o *speed up* do algoritmo.

2. Escreva um algoritmo BSP/CGM $n!$ com p processadores. Faça uma analise da complexidade do algoritmo incluindo o *speed up*.

3. Desenvolva um algorimo BSP/CGM (com p processadores) para verificar a paridade de um arquivo binário. Ou seja, verifique o número de 0's e 1's existentes no arquivo e adicione um 0 ou um 1 ao arquivo de tal forma que o número de 1's seja par.

4. Desenvolva um algoritmo para determinar o zero de uma função usando o método da bissecção. Neste método, dois pontos $f(a)$ e $f(b)$ são computados, (onde, digamos $f(a) > 0$ e $f(b) < 0$) então a função f deve interceptar o eixo x em algum ponto entre $f(a)$ e $f(b)$.

[*] Este material é para o uso exclusivo do Curso de Algoritmos Paralelos do DCT da UFMS. Parte desse material refere-se aos livros Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers de Barry Wilkinson e Michael Allen, Parallel Programming with MPI de Peter Pacheco, além de Exercicios elaborados por Edson Norberto Cáceres

5. Um determinado banco opera nacionalmente e recebe 30 milhões de cheques dos seus 2 milhões de clientes. Um dos problemas que o banco enfrenta é o de ordenar os cheques dos clientes de tal forma que ele possa efetuar os créditos e débitos correspondentes nas contas dos clientes e efetuar e compensar os cheques recebidos dos clientes que pertencem a outros bancos. O banco vem usando um computador de grande porte bastante rápido com um programa que faz a ordenação baseado no *quicksort*.Contudo, você conversou com o diretor do banco a disse que você conhece uma forma de fazer essa ordenação usando um conjunto de p estações de trabalho, as quais podem ordenar $1/p$ dos 30 milhões de cheques e intercalando os resultados parciais em um único resultado ordenado.Em função disso, você foi contratado como consultor do banco, para fazer uma simulação do processo utilizado para a ordenação dos cheques utilizando a programação com troca de mensagens.

   Suposições:

   - Cada cheque possui três números de identificação: dois números com nove dígitos (identificação do banco e identificação do cliente), e um número com três dígitos identificando o cheque (zeros a direita não são considerados).
   - Todos os cheques com a mesma identificação do banco são ordenados por cliente para serem encaminhados ao banco do cliente.
   - Estime o *speedup* para os casos em que $p = 10$, $p = 100$ e $p = 1000$. Estime a porcentagem de tempo gasto em comunicação versus o tempo gasto em computação.

6. Escreva um programa BSP/CGM, usando p processadores, para computar o valor do polinômio

$$f = a_0x^0 + a_1x^1 + a_2x^2 + \dots a_{n-1}x^{n-1}$$

   para qualquer grau n, onde os a's, x e n são as entradas. Analise a complexidade.

7. No algoritmo de ordenação por inserção, os conjuntos de números que são inseridos no *pipeline* possuem n/p elementos. Avalie o que pode ocorrer se inserirmos $\alpha\frac{n}{p}$ números, $\alpha < 1$, a cada rodada de comunicação.

8. Utilizando o crivo de Eratóstenes, desenvolva um algoritmo BSP/CGM, com p processadores, para computar o subconjunto de números primos existentes um conjunto dado S. Analise a complexidade do algoritmo.

9. Descreva um algoritmo BSP/CGM com p processadores para implementar a ordenação por rank. Analise a complexidade.

10. Na ordenação por bolha crescente, o maior número é movimentado para o final da lista através de uma série de comparações e trocas, comparações e trocas estas que começam a ser efetuados no início da lista. Ou seja, dados os números $x_0, x_1, \ldots, x_{n-1}$, inicialmente x_0 e x_1 são comparados e o maior é movimentado para x_1 (e o menor para x_0). Então x_1 e x_2 são comparados e o maior é movimentado para x_2, e assim sucessivamente até que o maior número esteja armazenado em x_{n-1}. Esse procedimento é repetido, até que o próximo maior número em $x_0, x_1, \ldots, x_{n-1}$ seja posicionado em x_{n-2}. Isso é repetido para cada número $x_{n-3}, \ldots, x_0$. Desta forma, os números maiores movem-se como uma *bolha* para o final da lista. Analise a complexidade do algoritmo seqüencial acima e projete e analise uma versão da ordenação por bolha para o modelo BSP/CGM com p processadores.