ALGORITMOS SISTÓLICOS PARA EXAMINAR PARES DE UM CONJUNTO

Li Kuan Ching

Orientador: Siang Wun Song

O problema de examinar pares de elementos é definido da seguinte forma: dado um conjunto de N elementos. examinar todos os pares deste. Suponhamos que os elementos sejam $1, 2, 3, \ldots, N$. Examinar os pares significa verificar

$$\begin{array}{llll} (1,2), & (1,3), & (1,4), & \ldots, \quad (1,N). \\ & (2,3), & (2,4), & \ldots, \quad (2,N). \\ & & (3,4), & \ldots, \quad (3,N). \\ & & & \vdots \\ & & & (N-1,N). \end{array}$$

Examinar pares de elementos pode ser necessário para a resolução de vários problemas. como por exemplo o problema da determinação de intersecção de retas, que pode ser enunciado como: Dados N segmentos de reta no espaço bidimensional Euclidiano. achar todos os pontos de intersecção dessas N retas. Um método de solucionarmos este problema é examinar todos os pares de segmentos de retas.

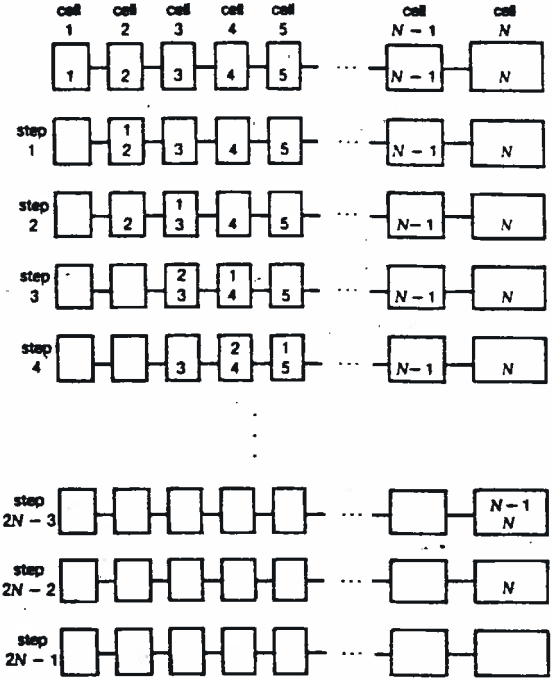
Apresentaremos a seguir quatro algoritmos para examinarmos pares de elementos. nas quais os dois primeiros são baseados na operação "Fold-Over" e os outros dois são adaptados de algoritmos de ordenação paralela.

Método 1

A idéia é tomarmos a lista do seu lado esquerdo e a dobrarmos. puxando-a da esquerda para a direita.

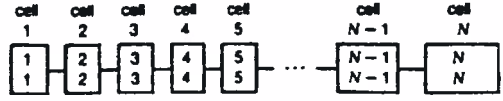
Note-se que o número de passos necessários para executarmos esta operação é $2N-2$. não levando em conta os N passos necessários para carregarmos cada elemento na sua respectiva célula. Se contabilizarmos também o tempo para a entrada. totalizará em $3N-2$

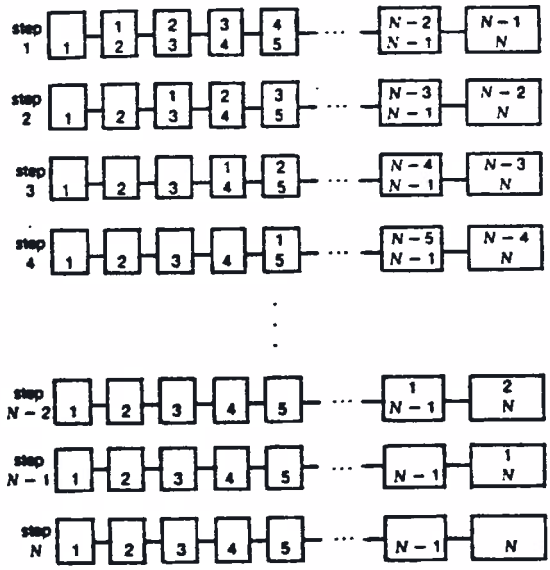
passos.



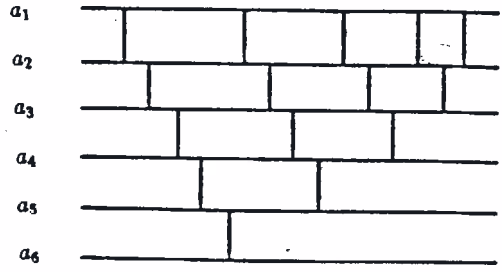
Método 2

Este que apresentaremos a seguir é um algoritmo melhorado a partir do anterior. Consiste em duplicarmos os dados durante a fase de inicialização, ou seja, existem duas cópias de i em cada célula i. Durante a operação, os elementos de baixo da célula i se mantêm na mesma posição, enquanto os de cima locomovem-se para a direita. O número total de passos necessários para completarmos o processo é de $N-1$. Se incluirmos o número de passos da fase de entrada, totalizar-se-ão $2N-1$ passos.



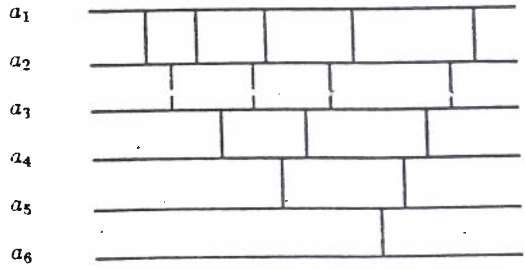


Em alguns algoritmos de ordenação, tais como "Bubblesort", Inserção Direta e Seleção Direta, todos os pares de dados serão comparados no pior caso. Para exemplificarmos a afirmação acima, a ordenação no algoritmo da Seleção Direta de 6 elementos é dada abaixo. Na figura, toda linha vertical indica operação de comparação e troca, se este último for necessário, ou seja, se $a_i > a_{i+1}$. Isto forçará para que o maior número seja escolhido após execução de cinco passos do algoritmo. Após os próximos quatro passos, determinaremos o segundo maior elemento, e assim continua.

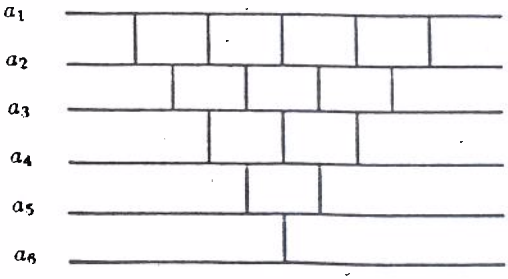


Note que, na figura acima, supusemos o pior caso, isto é, onde é necessária troca após cada comparação entre a_i e a_j, $i \neq j$, $1 \leq i < j \leq N$.

A figura a seguir mostra o algoritmo de Inserção Direta. Novamente, observamos que, no pior caso, todos os pares a_i e a_j serão examinados.



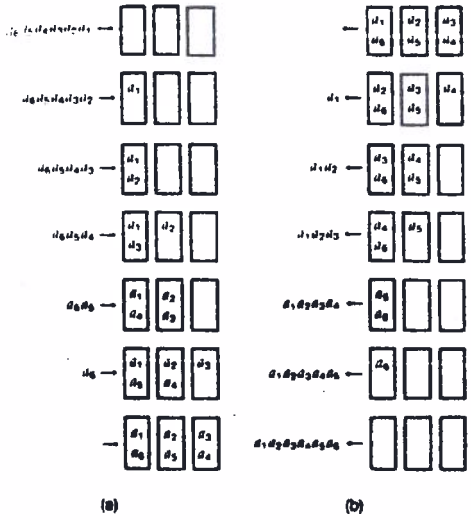
Vários algoritmos de ordenação paralela são baseados na observação do fato de que algumas comparações e trocas podem ser feitas simultaneamente. Para exemplificar, temos abaixo uma versão paralela do algoritmo de Inserção Direta. Da figura, temos que o número de comparações e trocas não diminuiu, mas sim o número de passos.



Método 3 - *Algoritmo baseado no Zero-Time Sorter*

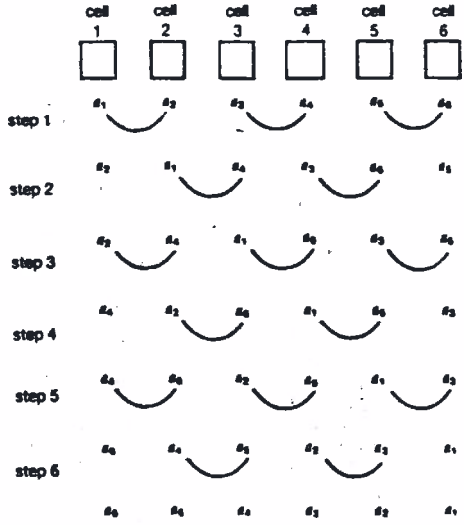
Originalmente, este algoritmo foi desenvolvido para fins de ordenação, mas a sua idéia pode ser aplicada ao nosso objetivo.

Este é chamado de "Zero-Time Sorter", pois o tempo de E/S sobrepõe completamente o tempo de computação. A operação pode ser dividida em duas fases: a de entrada e a de saída. Na fase de entrada, fig. (a), os elementos entram um por um, passando os maiores da esquerda para a direita, lembrando que é possível manter dois dados em cada célula. Enquanto que, na fase de saída, fig. (b), os dados também saem um por um e os menores passam da direita para a esquerda.



Método 4

Este é baseado no algoritmo de ordenação Par-Ímpar. Neste caso, os dados são inicialmente carregados. Nos passos ímpares, os dados das células $P_1, P_3, \ldots P_{N-1}$ são comparados com os das células $P_2, P_4, \ldots, P_N$ respectivamente, ocorrendo troca se for necessário. Nos passos pares, as células $P_2, P_4, \ldots, P_{N-2}$ são comparadas com os dados das células $P_3, P_5, \ldots, P_{N-1}$ respectivamente. Novamente, ocorrem trocas se for necessário. No pior caso, leva N passos para completar a ordenação. Na figura abaixo, os dados estão a princípio na ordem reversa. Conseqüentemente, é necessária a comparação de todos os pares de elementos do conjunto.



Para concluirmos, podemos dizer que se um algoritmo de ordenação compara todos os elementos do conjunto dado no seu pior caso, então esse algoritmo pode ser usado para examinarmos pares de elementos de um conjunto após pequena modificação.

Referência

[1] Akl, Selim G., *Parallel Sorting Algorithms*, Academic Press. 1985.

[2] Knuth D.E., *The Art of Computer Programming*, vol.3, Addison-Wesley, 1972.

[3] Song, S.W., *Algoritmos Paralelos e Arquitetura VLSI*, IV Escola de Computação, São Paulo, 1984.

[4] Zen-Cheung Shih, Gen-Huey Chen and R.C.T. Lee, *Systolic Algorithms to examine all Pairs of Elements*. pp.161-167. Communications of the ACM. February 1987.