

COSC329 Tutorial Problems on Algorithms
Preparation for parallel algorithms
August 2001

(1) The distance matrix multiplication of A and B is defined by $C = A * B$, where

$$c[i, j] = \min\{a[i, k] + b[k, j] \mid k=1, \dots, n\}.$$

If we correspond min to + and + to *, we have, similarly to ordinary matrices,

$$c[i, j] = a[i, 1] * b[1, j] + \dots + a[i, n] * a[n, j].$$

The closure of distance matrix A, A^* , is defined by

$$A^* = I + A + A^2 + \dots = I + A + A^2 + \dots + A^{n-1}.$$

Here the (i, j) element of identity matrix I is 0 if $i=j$ and infinity otherwise. Addition of matrices is done componentwise. This A^* gives all pairs shortest distances, since the (i, j) element of A^k gives the shortest distance from i to j that uses exactly k edges. This A^* is also computed as $A^* = (I + A)^{n-1}$. If we use repeated squaring, we can compute A^* by $\log n$ distance matrix multiplications, and the total time is $O(n^3 \log n)$.

Now compute A^* of the matrix in the last page of “Parallel Algorithms” by this method, sequentially first, and in parallel using the mesh approach in page 7.

(2) Develop a similar theory for reflexive-transitive closure and trace your method with a graph of your choice with 6 or more vertices.

(3) Trace mesh Floyd for shortest distances with the following matrix. Confirm the result by drawing the picture of the graph.

0	1	4	8	20
30	0	2	99	10
15	99	0	3	9
99	12	99	0	4
5	7	11	14	0

(4) Arrange n registers $a[i]$ horizontally with indices $i=1, \dots, n$. Each can hold a number. Let us sort these numbers in the following way.

Repeat the following $n/2$ times (n even), $(n+1)/2$ times (n odd)
 Compare and exchange $a[i]$ and $a[i+1]$ for $i=1, 3, \dots$ in parallel
 Compare and exchange $a[i]$ and $a[i+1]$ for $i=2, 4, \dots$ in parallel

Trace this algorithm for (21 24 36 19 41 64 69 45 76 11)

(5) Develop a parallel algorithm that computes the prefix minimum for each array position in $O(\log n)$ time. Here the prefix minimum of the i-th position is minimum of $a[1], \dots, a[i]$. Trace this algorithm with the sequence given in (4). Is $O(n)$ cost possible?