

## Assignment 4.

Input: D0,  $n \times n$  matrix with 0 on diagonal,  
positive values other places

Output: D,  $n \times n$  matrix

```
for k starting from 0 through n-1
  for i starting from 0 through n-1
    for j starting from 0 through n-1
      D[i][j] = min{ D0[i][j], D0[i][k]+ D0[k][j] }
    end for j
  end for i

  for i starting from 0 through n-1
    for j starting from 0 through n-1
      D0[i][j] = D[i][j]
    end for j
  end for i
end for k
```

Write an MPI code that parallelizes the pseudo code given above by parallelizing the inner for-i and for-j loops in a way that each process can allocate memories only for a sub-matrix of D and a sub-matrix of D0 of size  $n/\sqrt{P} \times n/\sqrt{P}$ , where  $P$  is the number of processes. You may allocate up to 4 more 1-D arrays of size  $n/\sqrt{P}$ . You may assign processes as in the figure below to make the parallelization easier, where the number in each small square is the process id, i.e. the rank of the process. You may assume  $n$  is divisible by  $\sqrt{P}$  and  $P$  is the square of a positive integer. The communication cost of your parallel algorithm should be bounded by  $O\left(\frac{n}{\sqrt{P}} \log_2 \sqrt{P}\right)$  for each k-loop iteration.

0	1	...	$\sqrt{P}-1$
$\sqrt{P}$	$\sqrt{P}+1$	...	$2\sqrt{P}-1$
...	...	...	...
??	??+1	...	$P-1$