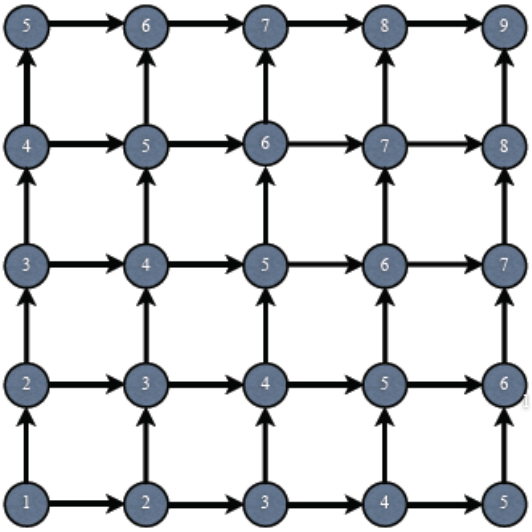
# Homework 10

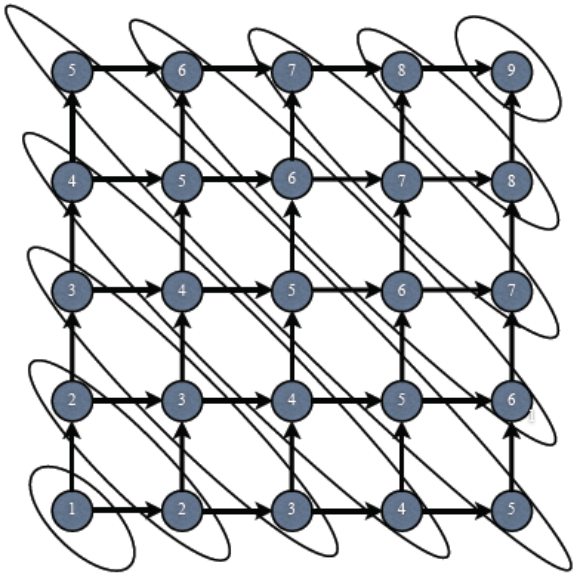
**Explanation:**

Consider the loop:

```
for (i=0; i < n; i++) {
  for (j=0; j < n; j++) {
    a[i][j] = a[i-1][j] + a[i][j-1];
  }
}
```

This loop is not fully parallel because of dependences on the *i* and *j* loops. A diagram of the dependences is shown on the left, and on the right collections of nodes that can execute together




are shown. The numbers within the nodes indicate at what step in the loop a node can execute.

This sort of parallelism is called *wavefront* parallelism because the iterations that can execute in parallel moves like a wave through the full iteration space. The node labeled ***1*** executes first, which fulfills the dependences for the nodes labeled ***2***, which fulfills the dependences for the nodes labeled ***3***, and so forth.

**Question 1:** Let one element of the array be updated by each process, i.e. the processor labeled $P_{i,j}$ updates *a[i][j]* in the iteration space.

**(a)** What is the work for the entire computation?
**(b)** What is the work at each node?
**(c)** What is the average degree of concurrency in the algorithm, i.e. how many processors can be executing in parallel performing the computation of the loop? You can assume a square matrix.
**(d)** Assume communication takes 1 unit of time, an iteration of the loop takes one unit of time, and a node can send two messages at once. Give an estimate of the sequential overhead, and an estimate of the maximum speedup predicted by Amdahl's law.
**(e)** What is the parallel overhead?
**(f)** Optional (this is very tricky) What is the Isoefficiency relation for this problem?