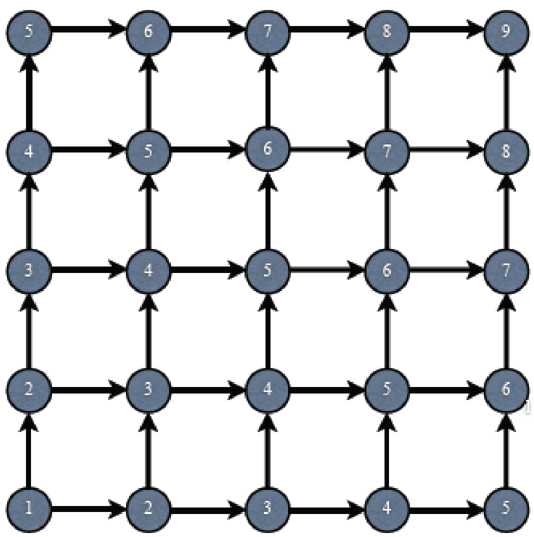
Homework 10

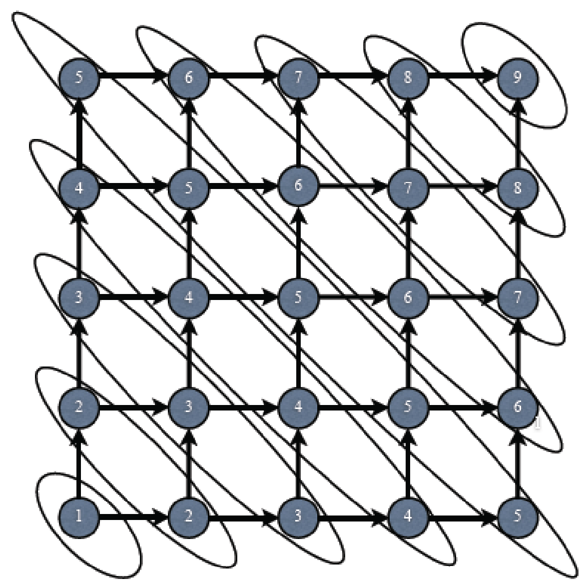
Explanation:

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?

n^2 -- this is the number of operations performed in the loop.

(b) What is the work at each node?

O(1), if there is one node per element.

(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.

There are $1, 2, 3, \ldots, \sqrt{P}$ nodes executing during each of the first $\sqrt{P}$ time steps, and then $\sqrt{P}-1, \sqrt{P}-2, \ldots, 1$ nodes executing during each of the next $\sqrt{P}-1$ time steps. The sum of these two series is $(\sqrt{P}(\sqrt{P}+1)+(\sqrt{P}-1)(\sqrt{P})) / 2$ using Gauss' solution. Multiplying this expression out gives us P processors active at some point during the $2\sqrt{P}-1$ time steps that the wavefront is executing. Since we are looking for the average, we simply divide P by $2\sqrt{P}-1$, which yields $P / (2\sqrt{P}-1)$. As P grows large, this becomes $\sqrt{P}/2$ average concurrency, or, dropping the constants, ($\sqrt{P}$) average concurrency.

(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~~ sequential fraction *f* (the s**erial path through the program), and an estimate of the maximum speedup predicted by Amdahl's law.**

This was not a particularly well worded question in the original homework. What I wanted was an estimate of the serial path through the program and the maximum parallelism. The serial path through the program is approximately $2\sqrt{P}$ long, (note that $n = \sqrt{P}$ in this problem), and the serial fraction of the work is $(2\sqrt{P})/n2$, or $2n/n^2$ or $2/n$, and the maximum speedup is $n/2$.

(e) What is the parallel overhead?

Communication requires O(2) communication operations per time step, and a total of $2\sqrt{n}-1$ time steps. All P processors are involved for all time steps (by involved I don't necessarily mean doing useful work, but they are part of the job.) and so $TO = P(4\sqrt{n}-1)$. See the supplement solution to this question to see a fuller answer.

(f) Optional (this is very tricky) What is the Isoefficiency relation for this problem?
See the supplement solution to this question for the answer.