Upcoming Schedule

- One more programming assignment, details likely Thursday after the break. Due date will be April 11, the last class meeting before final projects.
- Final Project presentations last 2 weeks of classes and ***maybe*** Finals week (Tue April 30 6pm).
- Proposals for projects will be due March 21.

Final Project preview

Proposal Due March 21

- Statement of the problem you want to solve.
- Outline of approach or algorithm you plan to use, References?
- Questions or issues do you plan to investigate.
- It want, you can discuss your idea with me beforehand.
- Undergrad projects can be a teamed project, no more than 3 students per team. Project scope should match team size.

Possibilities

- Numerical PDEs
- Numerical Linear Algebra
  - Sparse Matrix times Vector
- Monte Carlo Methods
  - Neutron Transport
- Sorting
- Others: searching, dense linear algebra, maximal independent sets, graph coloring …
- Any topic from texts that we haven't covered
- Other styles of parallelism on other machines: "Proof of concept" recommended before proposal.

Numerical PDEs

Parallel multigrid solvers for partial differential equations in coastal ocean simulation

Jim E. Jones: Math

Steven Jachec: Marine and Environmental Systems

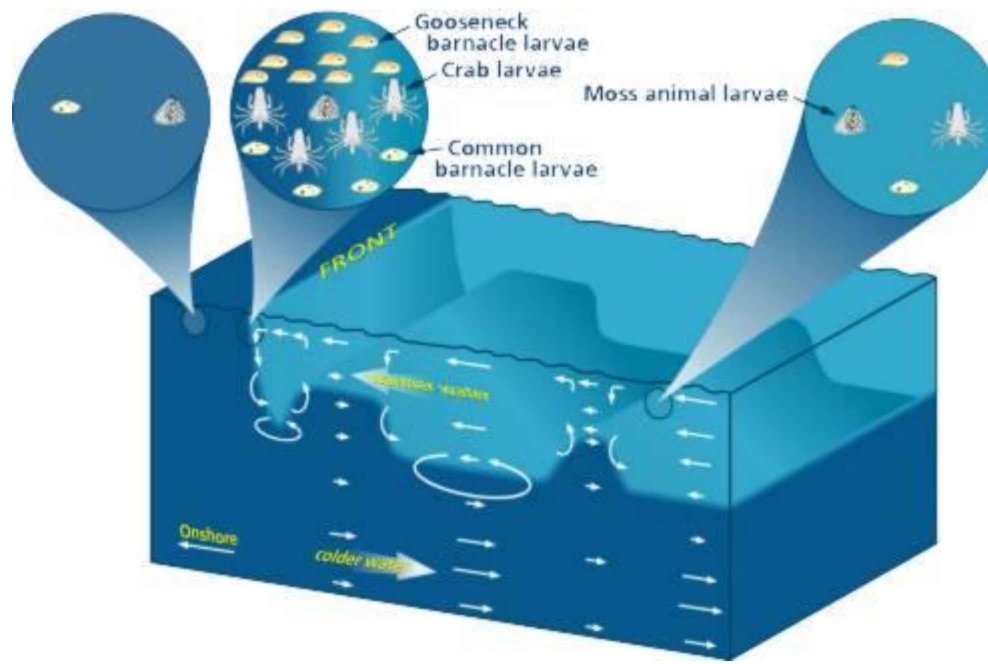
Anjali Ram PhD Math 2011

Osita Onyejekwe MS Math 2012

Work partially supported by NSF Grant No. CNS 09-23050
ONR Early Student Support Grant (# N000141110170)

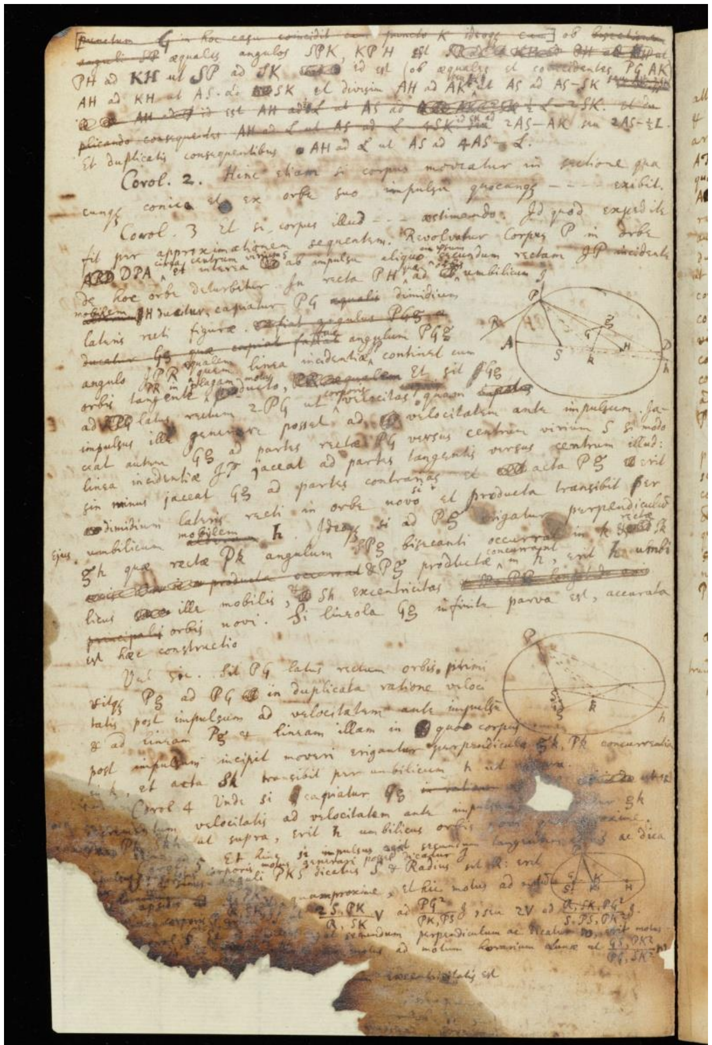
Tidal effects: surface and internal

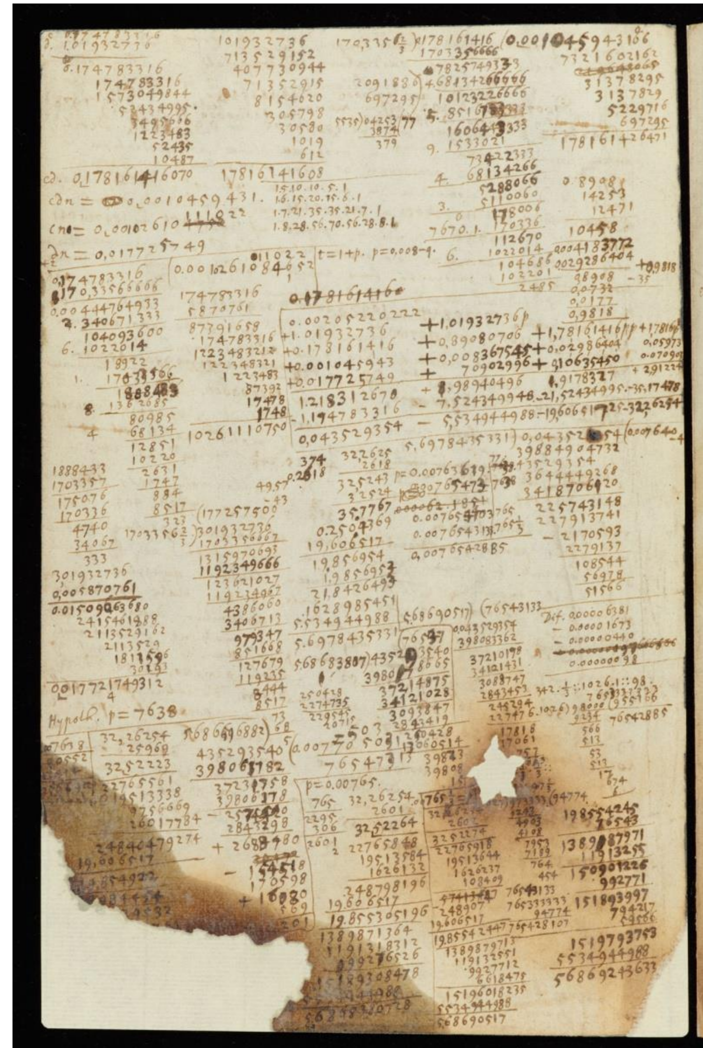
Bay of Fundy, New Brunswick, Canada



Pineda *Woods Hole*, Annual Report 1998

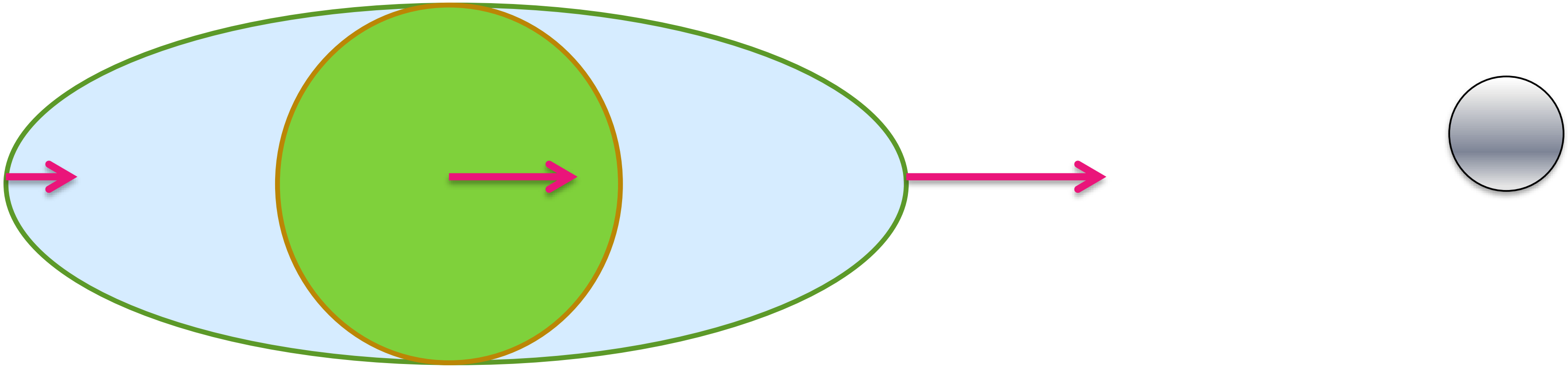
Newton's Principia
Early numerical model of tides





$$F = ma$$

$$F = G\frac{mM}{r^2}$$



Stanford Unstructured Non-hydrostatic Terrain-following Adaptive Navier-Stokes Simulator

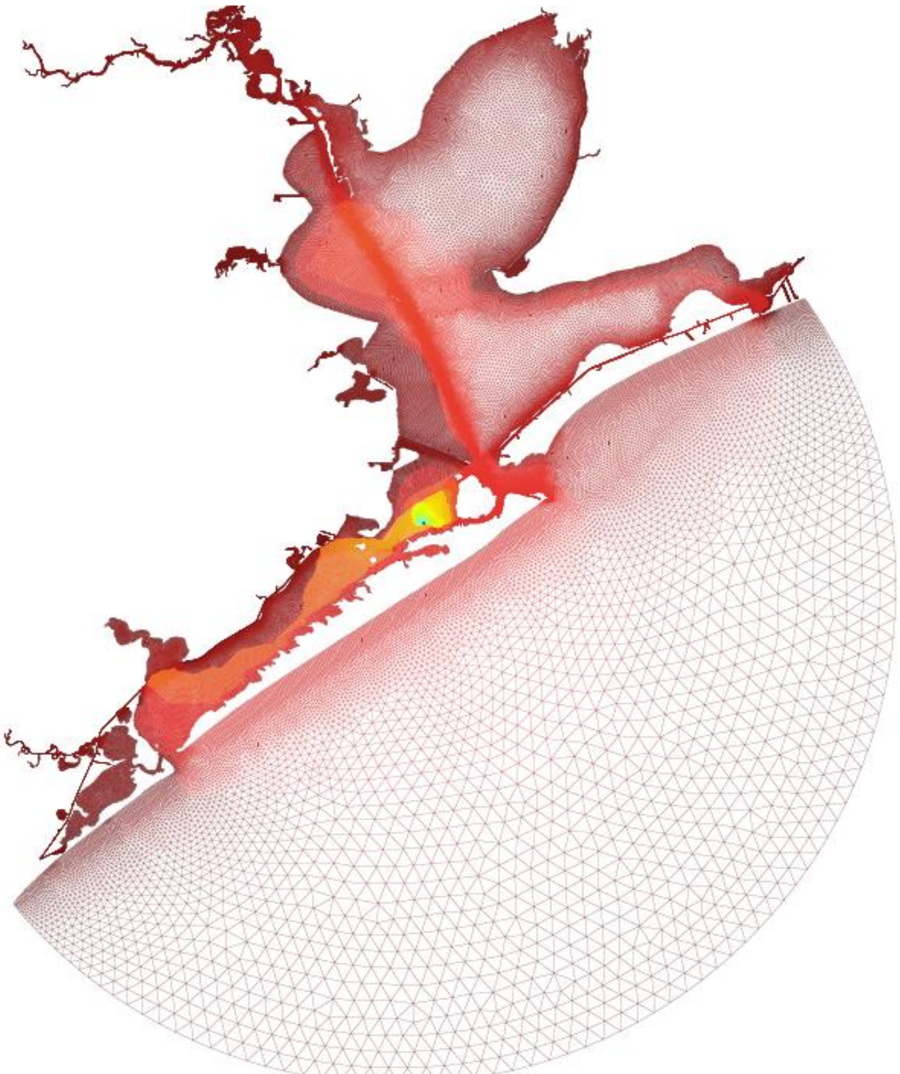
The three-dimensional Navier-Stokes equations with the Boussinesq approximation in a rotating frame, after filtering with either Reynolds-averaging or via a large-eddy simulation and employing an eddy viscosity model, are given by

$$\frac{\partial u}{\partial t}+\nabla\cdot(\vec{u}u)-fv+bw=-\frac{1}{\rho_0}\frac{\partial p}{\partial x}+\nabla_H\cdot(\nu_H\nabla_H u)+\frac{\partial}{\partial z}(\nu_V\frac{\partial u}{\partial z})$$

$$\frac{\partial v}{\partial t}+\nabla\cdot(\vec{u}v)+fu=-\frac{1}{\rho_0}\frac{\partial p}{\partial y}+\nabla_H\cdot(\nu_H\nabla_H v)+\frac{\partial}{\partial z}(\nu_V\frac{\partial v}{\partial z})$$

$$\frac{\partial w}{\partial t}+\nabla\cdot(\vec{u}w)-bu=-\frac{1}{\rho_0}\frac{\partial p}{\partial z}+\nabla_H\cdot(\nu_H\nabla_H w)+\frac{\partial}{\partial z}(\nu_V\frac{\partial w}{\partial z})-\frac{g}{\rho_0}(\rho_0+\rho)$$

Fringer, Gerriten, Street *Ocean Modeling* 2006



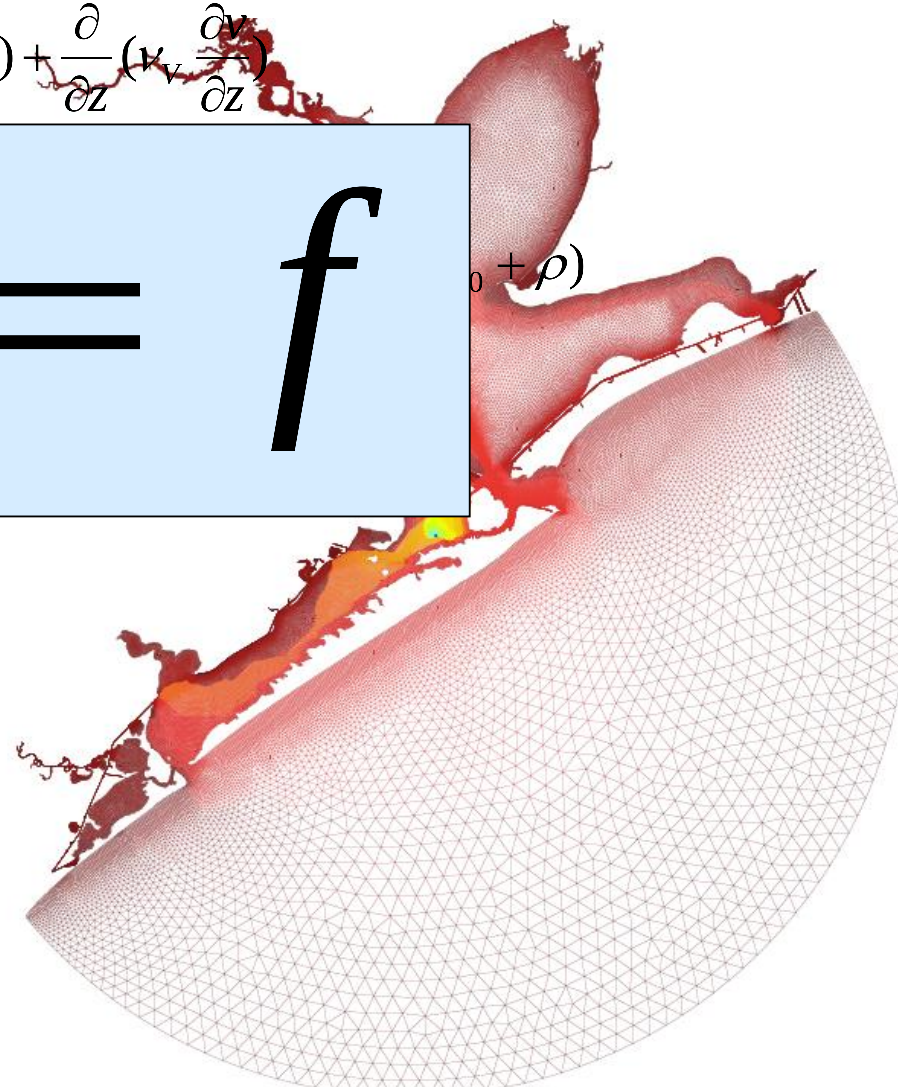
http://www.stanford.edu/~fringer/research.html

$$\frac{\partial u}{\partial t} + \nabla \cdot (\vec{u}u) - fv + bw = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} + \nabla_H \cdot (\nu_H \nabla_H u) + \frac{\partial}{\partial z}(\nu_V \frac{\partial u}{\partial z})$$

$$\frac{\partial v}{\partial t} + \nabla \cdot (\vec{u}v) + fu = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} + \nabla_H \cdot (\nu_H \nabla_H v) + \frac{\partial}{\partial z}(\nu_V \frac{\partial v}{\partial z})$$

$$\frac{\partial w}{\partial t} \quad \ldots \quad {}_0 + \rho)$$

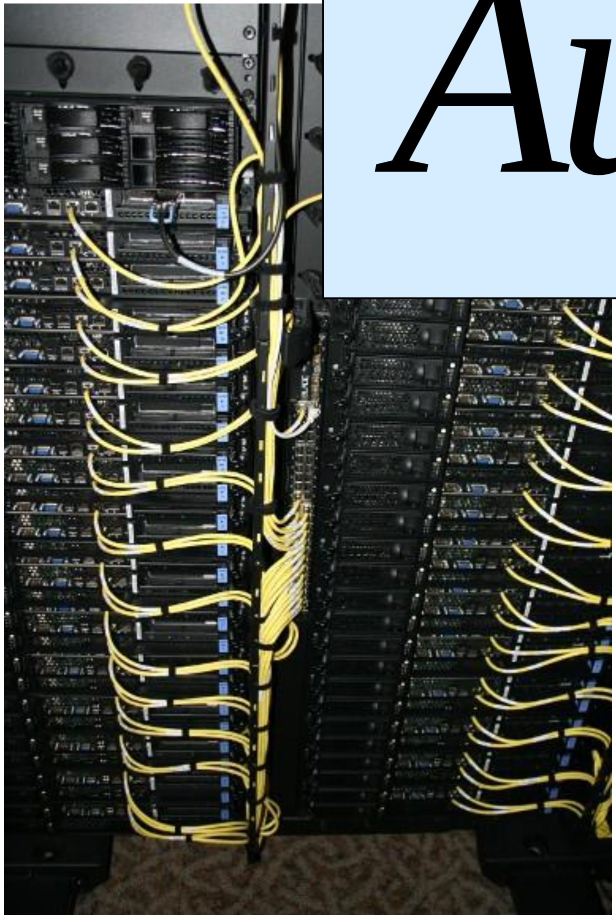
$$Au = f$$



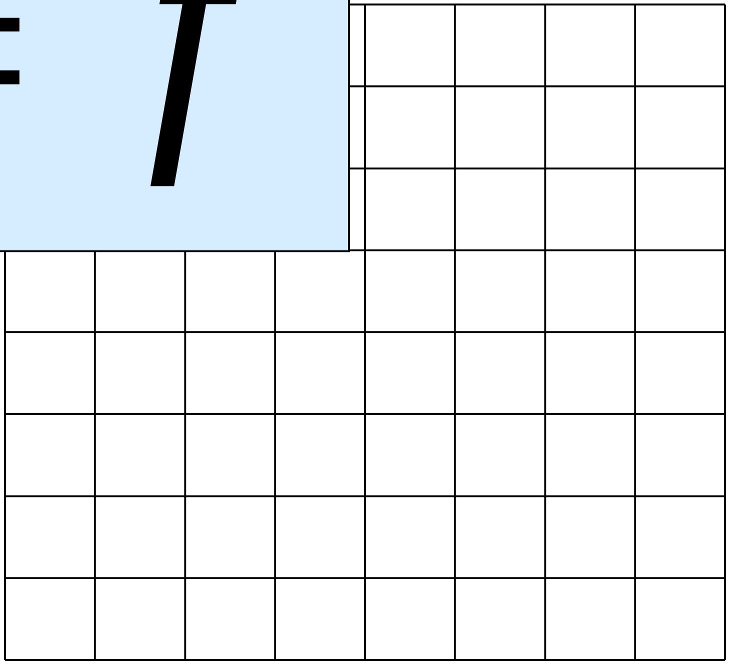
If you can’t do a problem, there is a simpler problem you do not understand. Your first job is to find that simpler problem.

Achi Brandt

$$u_t - (u_{xx} + u_{yy}) = f$$

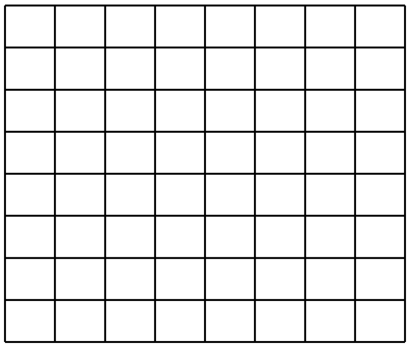


$$Au = f$$



Discretization approximates the differential problem by an algebraic one

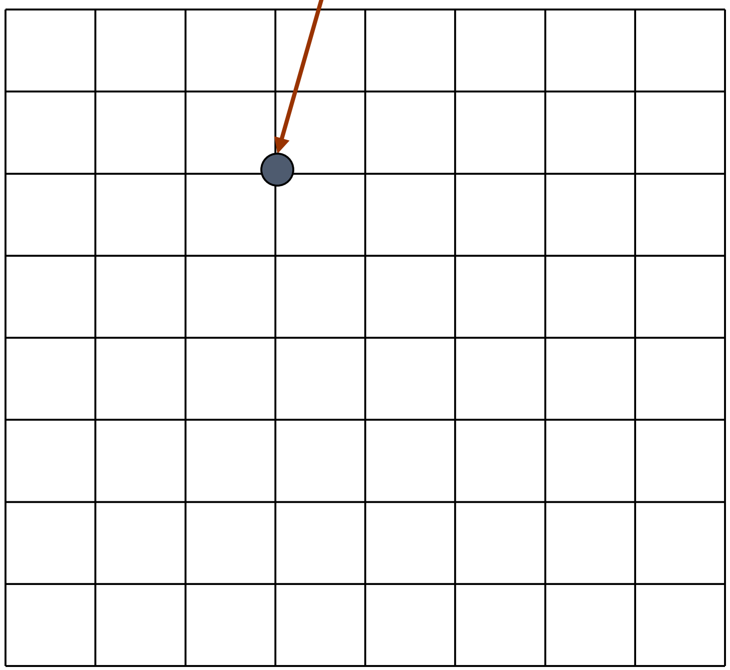
$$u_{xx} + u_{yy}$$



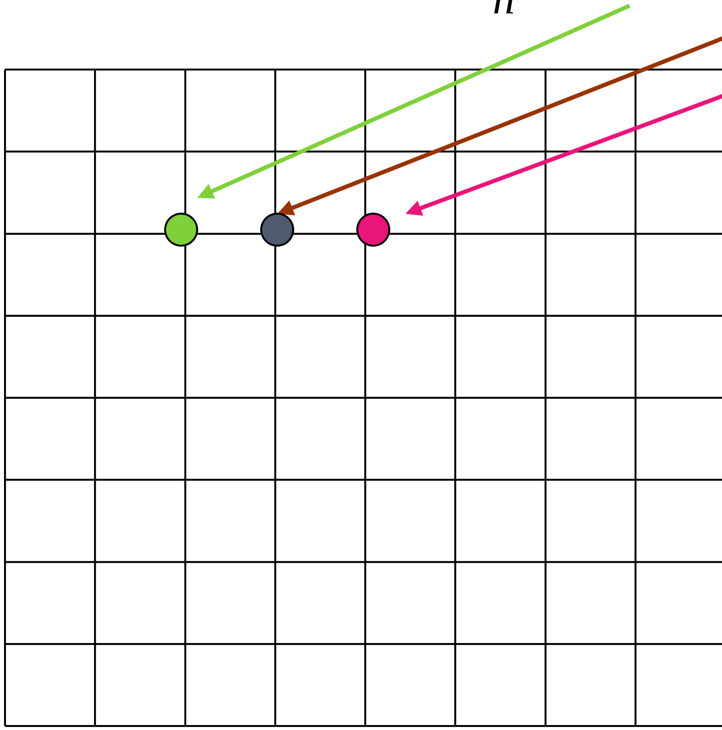
$$Au = f$$

Continuous to Discrete in 2D

$$u_{xx}(x,y) \approx \frac{1}{h^2}\left(u(x-h,y) - 2u(x,y) + u(x+h,y)\right)$$



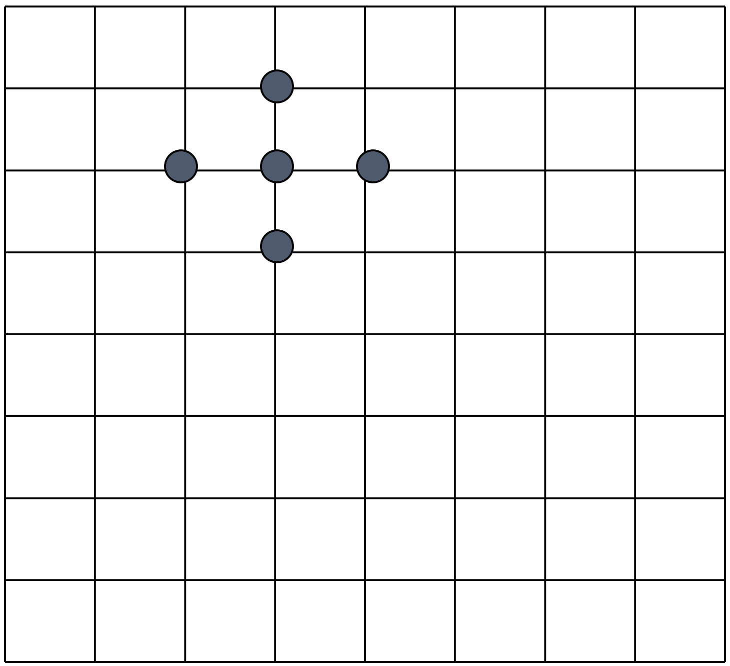
$$u_{xx}(x,y) \approx \frac{1}{h^2}(u(x-h,y) - 2u(x,y) + u(x+h,y))$$



Approximated derivative at a point by an algebraic equation involving function values at nearby points

By Taylor's Theorem, the error in this approximation (the truncation error) is ***O(h²)***

$$u_{xx} + u_{yy} \Rightarrow \frac{1}{h^2}(u(x-h,y)+u(x,y-h)-4u(x,y)+u(x+h,y)+u(x,y+h))$$



Error in approximation is determined by the mesh size ***h***. Difference between differential solution and algebraic solution goes to zero as ***h*** does.

Solve the heat equation in 2d

- Develop Parallel Code.
- Run on Blueshark.
- Investigate speed-ups and scalability.
- Explore different partition strategies.

Related PDE possible projects

- Implicit methods for time dependent problems.
- Use parallel libraries (hypre, PetSc, …)
- Run on Blueshark.
- Investigate speed-ups and scalability.

Sparse MPI Matvec

Sparse Matrix times Dense Vector

- y=Ax
- Store only non-zero's of A

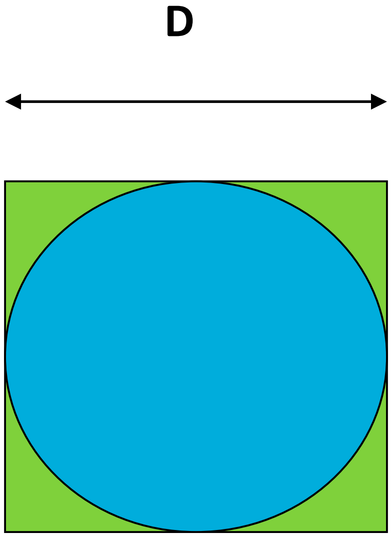
$$
\begin{bmatrix}
6 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 8 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 4 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 2 & 2 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 15 & 0 & 0 & 1 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 1 \\
2 & 0 & 0 & 0 & 0 & 1 & 0 & 8 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 \\
0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 37
\end{bmatrix}
\begin{bmatrix}
2 \\ 8 \\ 5 \\ 6 \\ 12 \\ 44 \\ 1 \\ 9 \\ 0 \\ 14
\end{bmatrix}
$$

Sparse Matrix times Dense Vector

- Develop Parallel MPI Code.
- Challenge is minimize the communication and allow general sparsity structure.
- Run on Blueshark.
- Investigate speed-ups and scalability.

Monte Carlo

Monte Carlo Methods



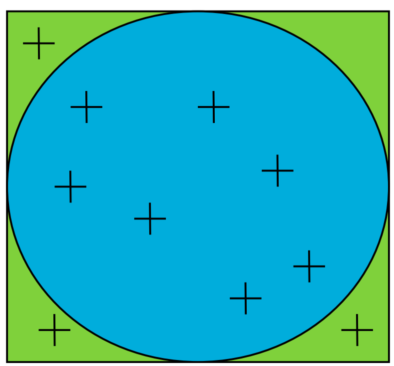
- Ratio of areas

$$\frac{\pi D^2 / 4}{D^2} = \frac{\pi}{4}$$

Monte Carlo Methods

D

$$\frac{\pi D^2/4}{D^2} = \frac{\pi}{4}$$



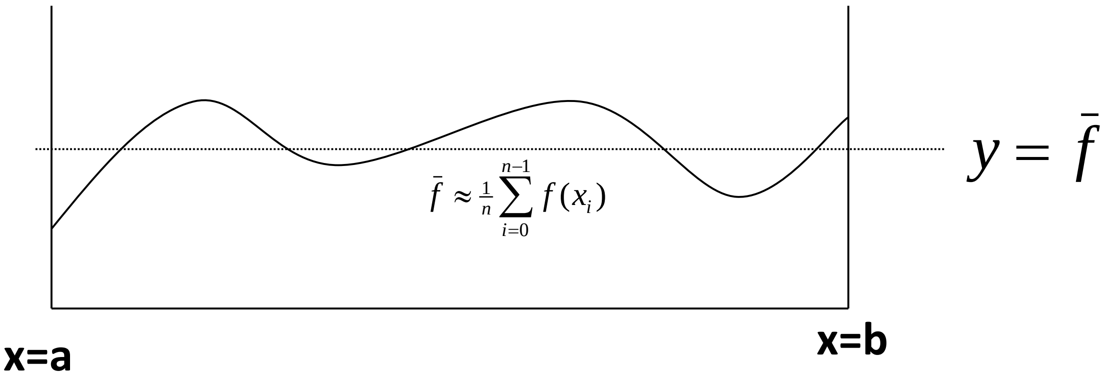
$$\frac{\pi}{4}$$

- Ratio of areas

- Ratio of “hits”

Monte Carlo Methods

$$I = \int_a^b f(x)dx = (b-a)\,\bar{f}$$



- Monte Carlo approximates
- Parallelization by running trials on each processor

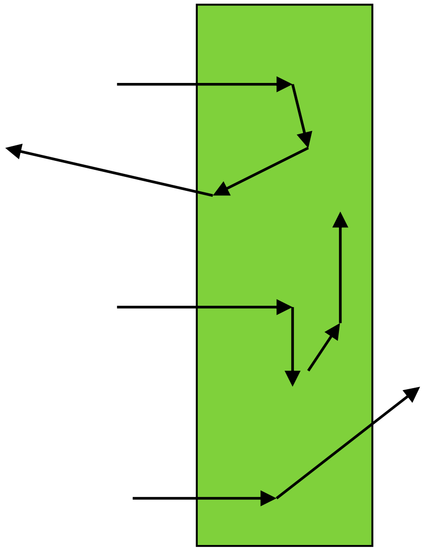
Monte Carlo Methods

- Issues
  - accuracy vs. number of trials
  - parallel random number generators
- Applications
  - Neutron Transport

Monte Carlo Methods

- Issues
  - accuracy vs. number of trials
  - parallel random number generators
- Applications
  - Neutron Transport

$C = C_s + C_c$



While not done

Travel Random Distance L=-ln(u)/C

If outside slab, done.

Interact with atom (random outcome)

If ($u<C_c/C$) absorbed, done.

Else scattered in random direction (d=2*pi*u)

Monte Carlo Methods

- Pick an application (Neutron transport?)
- Develop Parallel Code.
  - Will require mostly looking at parallel random number generation.
- Run on Blueshark.
- Investigate speed-ups and scalability.
- Investigate accuracy vs. number of trials

Sorting

Parallel Sorting

- Given a set of n integers on each processor

6	20	5	73

1	101	3	66

1	32	261	55

- Produce a sorted list: sorted within a processor and largest integer on p0 less than or equal to smallest on p1 when rank(p0)<rank(p1).

1	1	3	5

6	20	32	55

66	73	101	261

Parallel Sorting

- Quinn Ch 14 discusses Parallel Sorting.
- I can provide other references as well.
- Project might be
  - Implement one or more of the other parallel sorting algorithms
  - Compare to Sorting performance.
  - Compare to models.
- FYI: this is a popular project topic but is difficult to do well and see significant speedups.

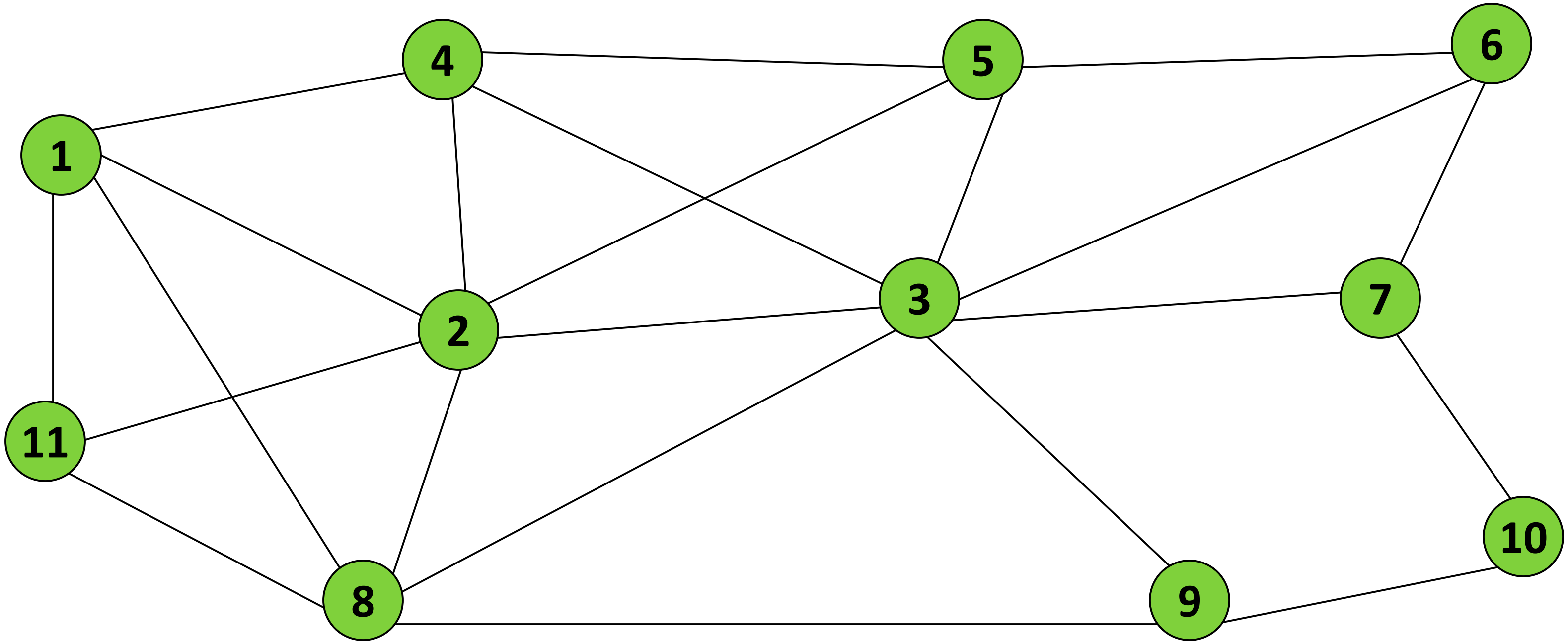
Worker Manager

- Certain applications will be well suited to this style:
  - Game trees
  - Number theory problems
  - Any problem where tasks vary in length in unpredictable ways
- Run on Blueshark.
- Investigate speed-ups and scalability.

Graph Coloring

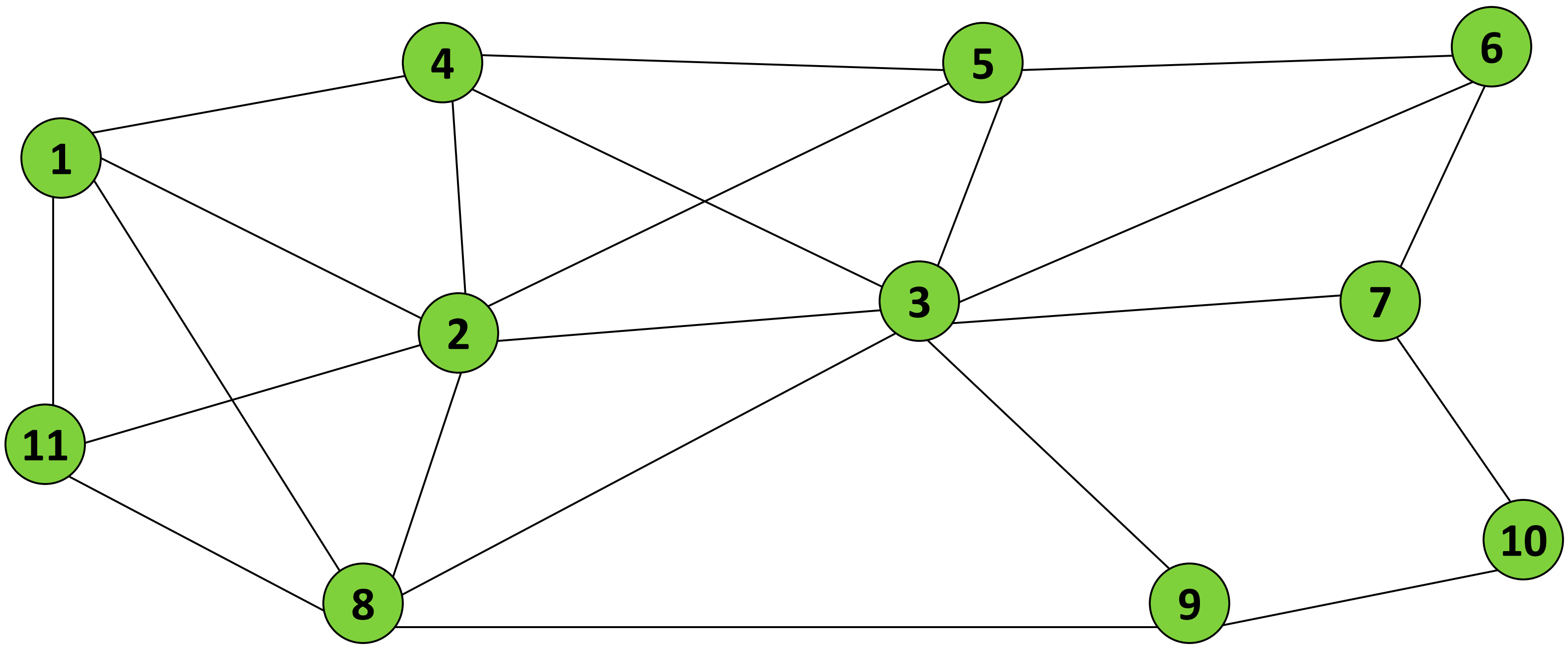
Graph Theory Definitions

- A **graph** $G=(V,E)$ is a collection of **vertices** $V=\{v_1,v_2, \dots , v_n\}$ connected by **edges** $E=\{e_1,e_2, \dots , e_m\}$.



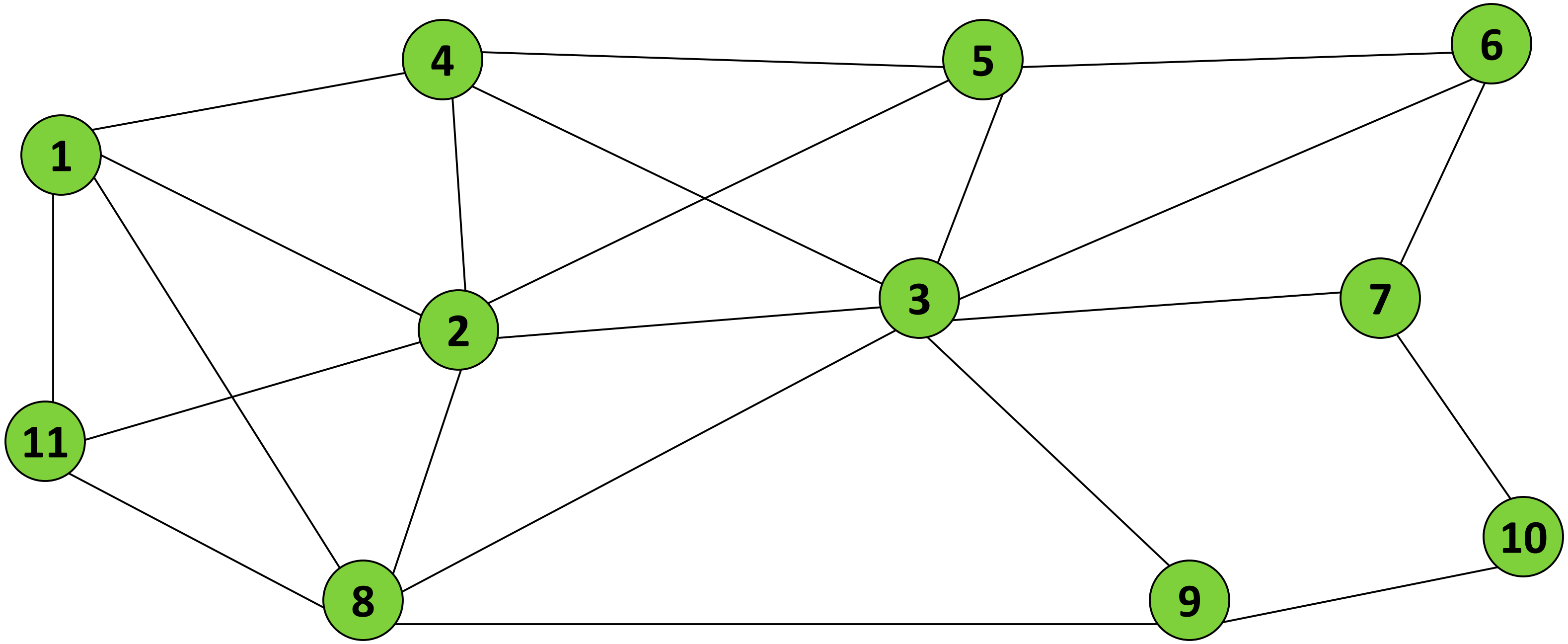
Independent Set Definition

- A set of vertices is a **independent set** if none of the vertices are connected by an edge



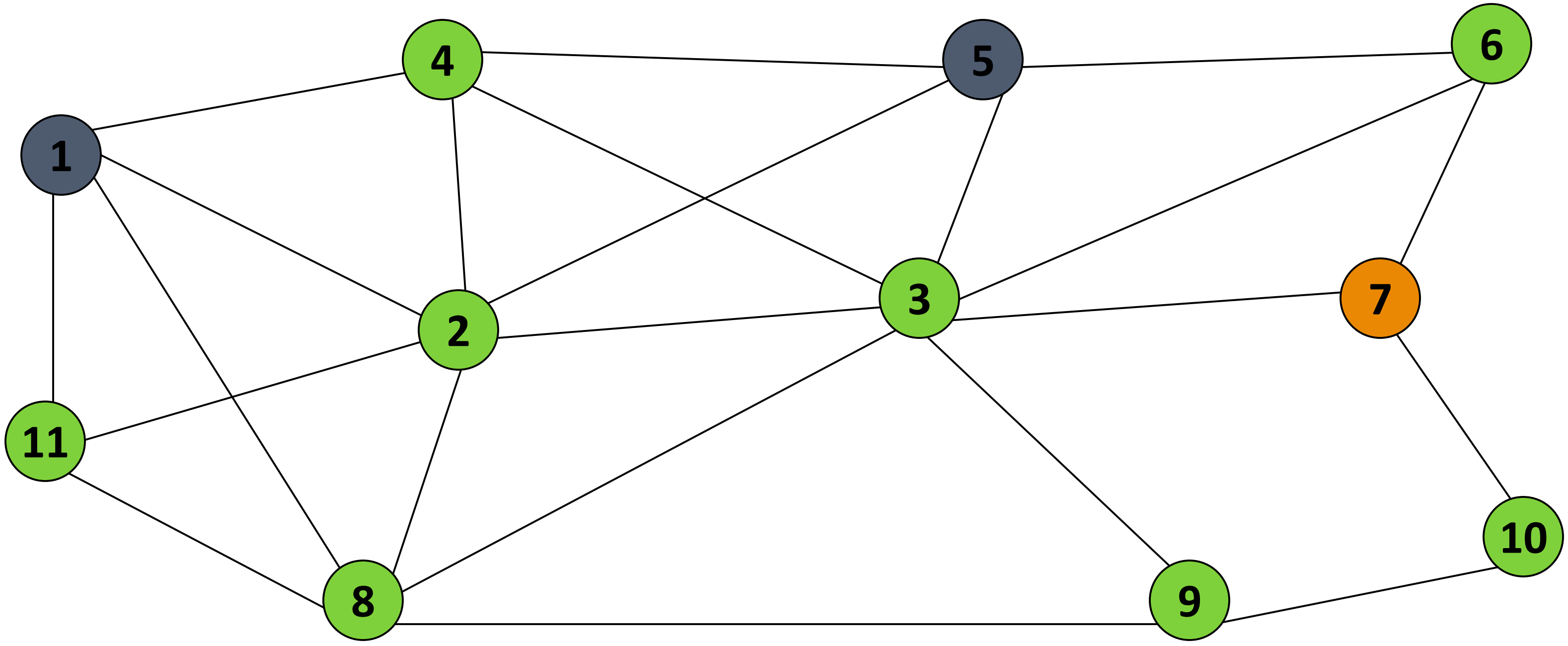
Maximal Independent Set Definition

- A set S of vertices is a **maximal independent set** (MIS) if it is an independent set and adding any other vertex to the set produces a set which is not independent.



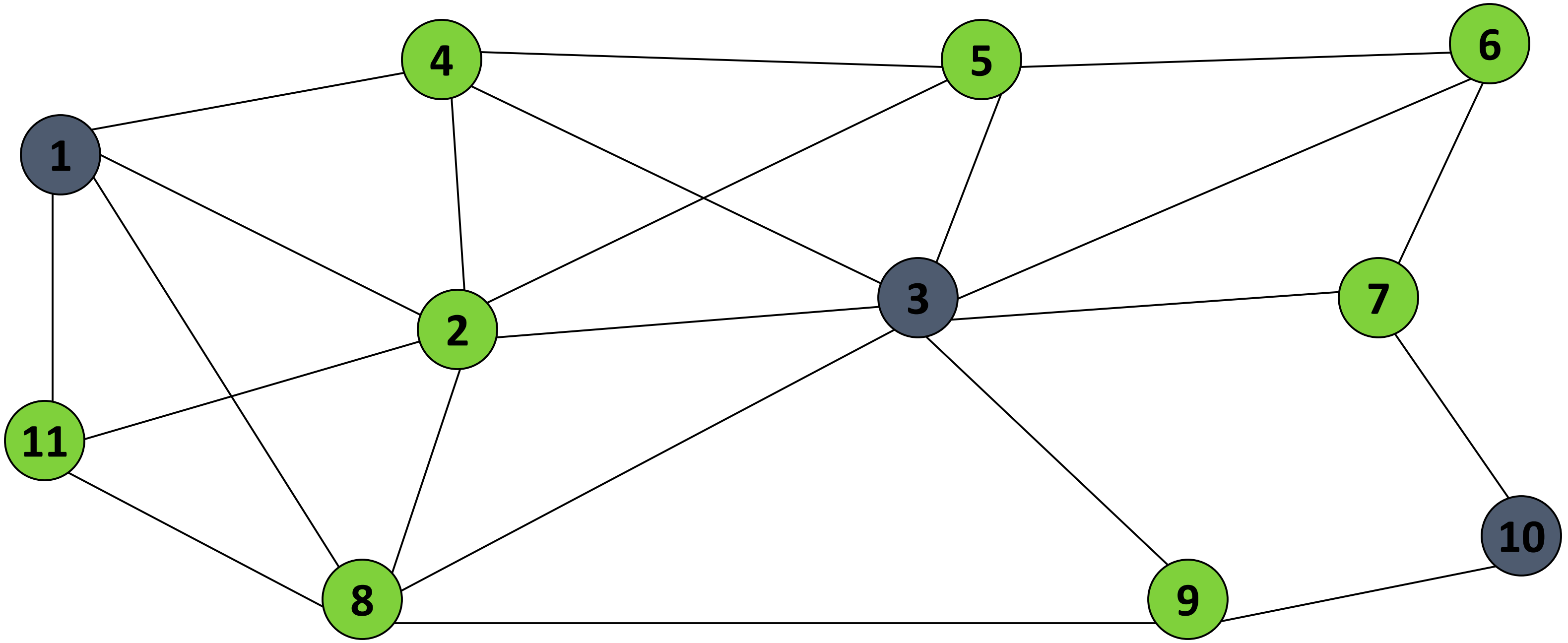
Independent Set Example

- S_1={1,5} is not a MIS because we could add, say, vertex 7 to it and still have a independent set.



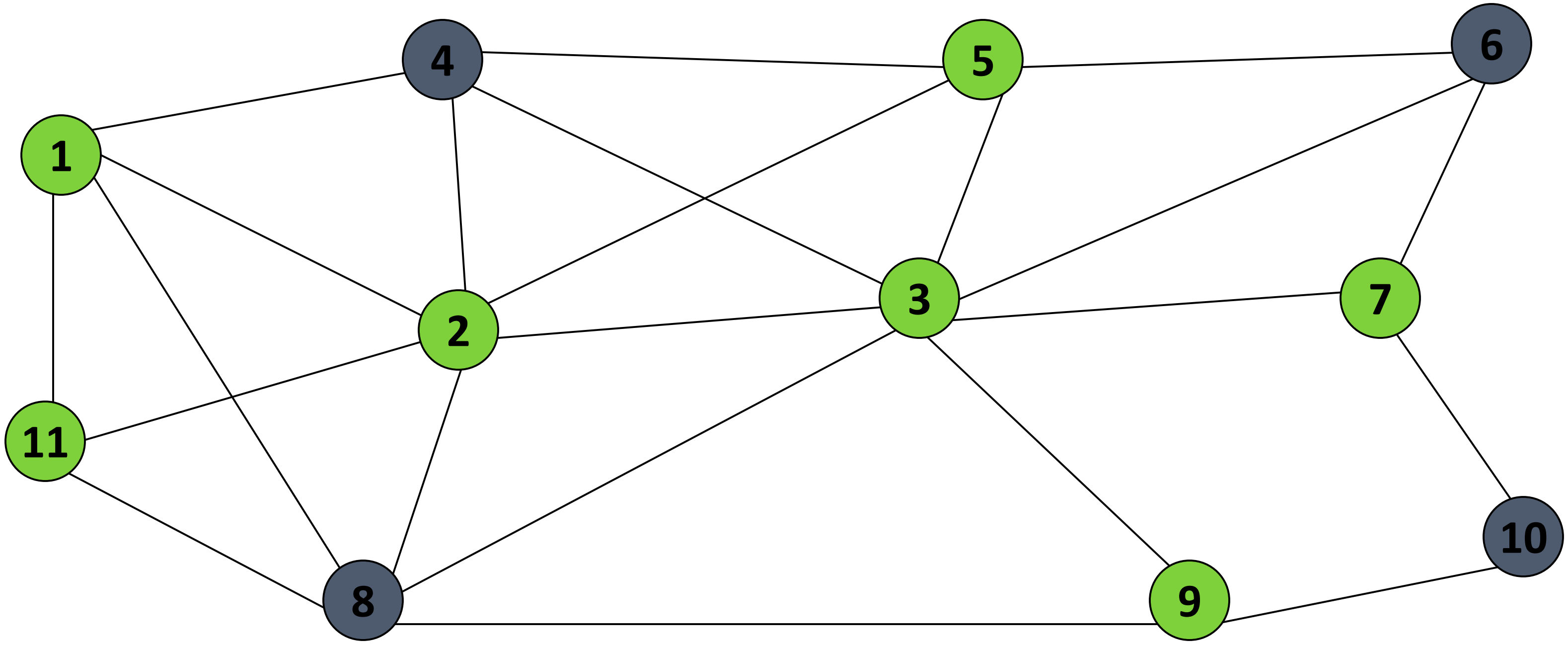
Maximal Independent Set Example

- S_2={1,3,10} is a MIS



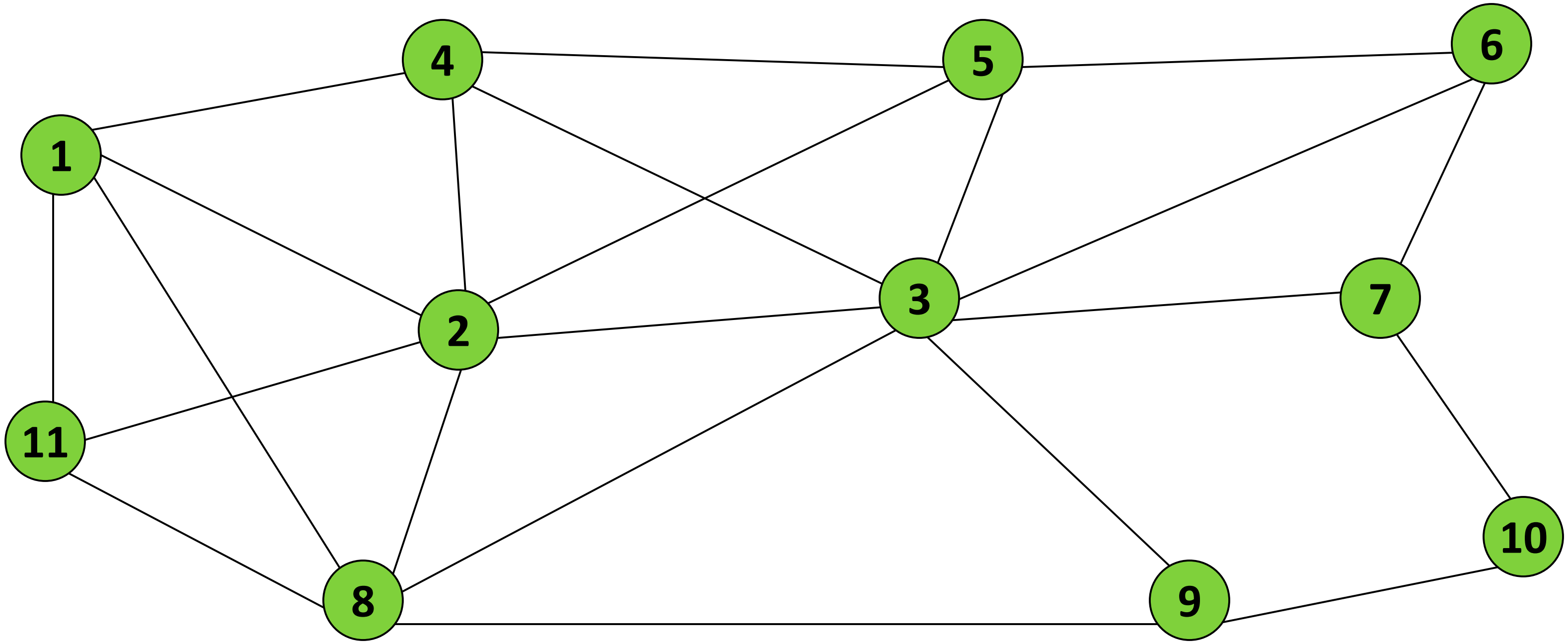
Maximal Independent Set is not unique

- S_3={4,6,8,10} is a MIS



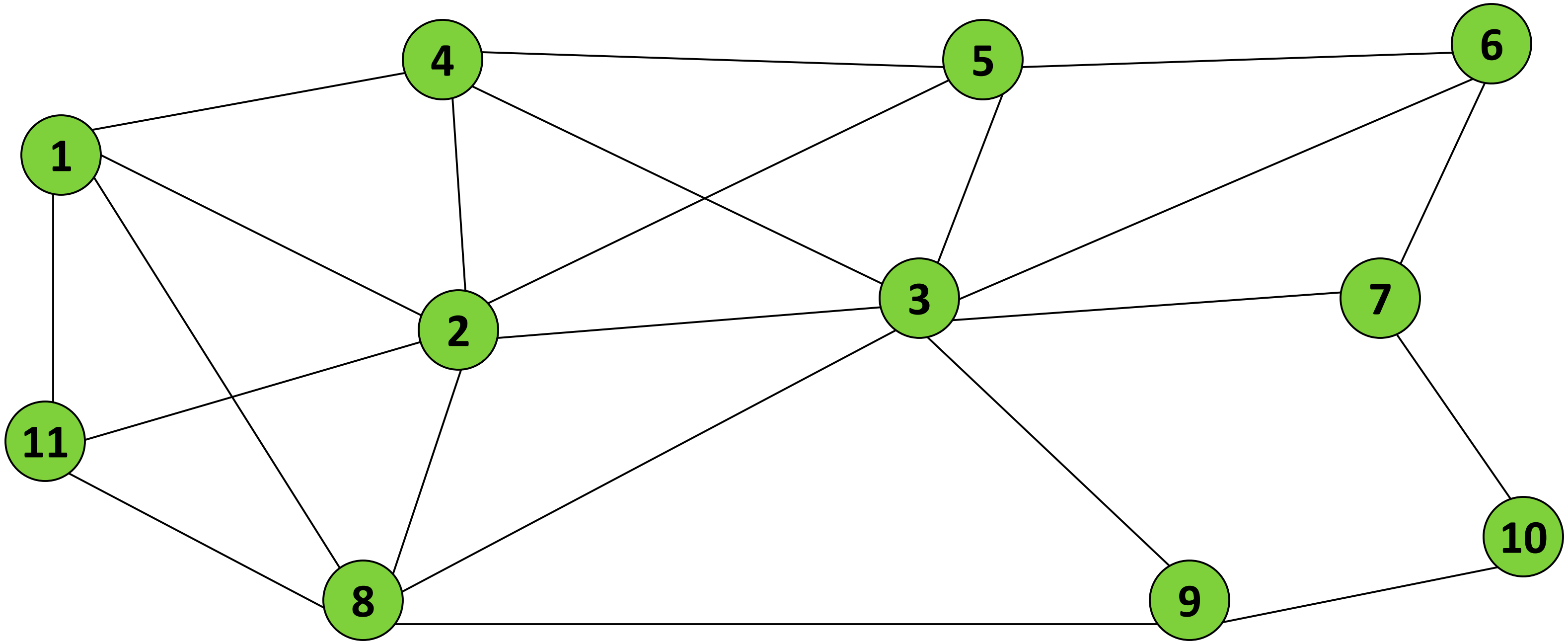
Maximal Independent Set Algorithm?

- Given a graph G=(V,E), find a maximal independent set S.



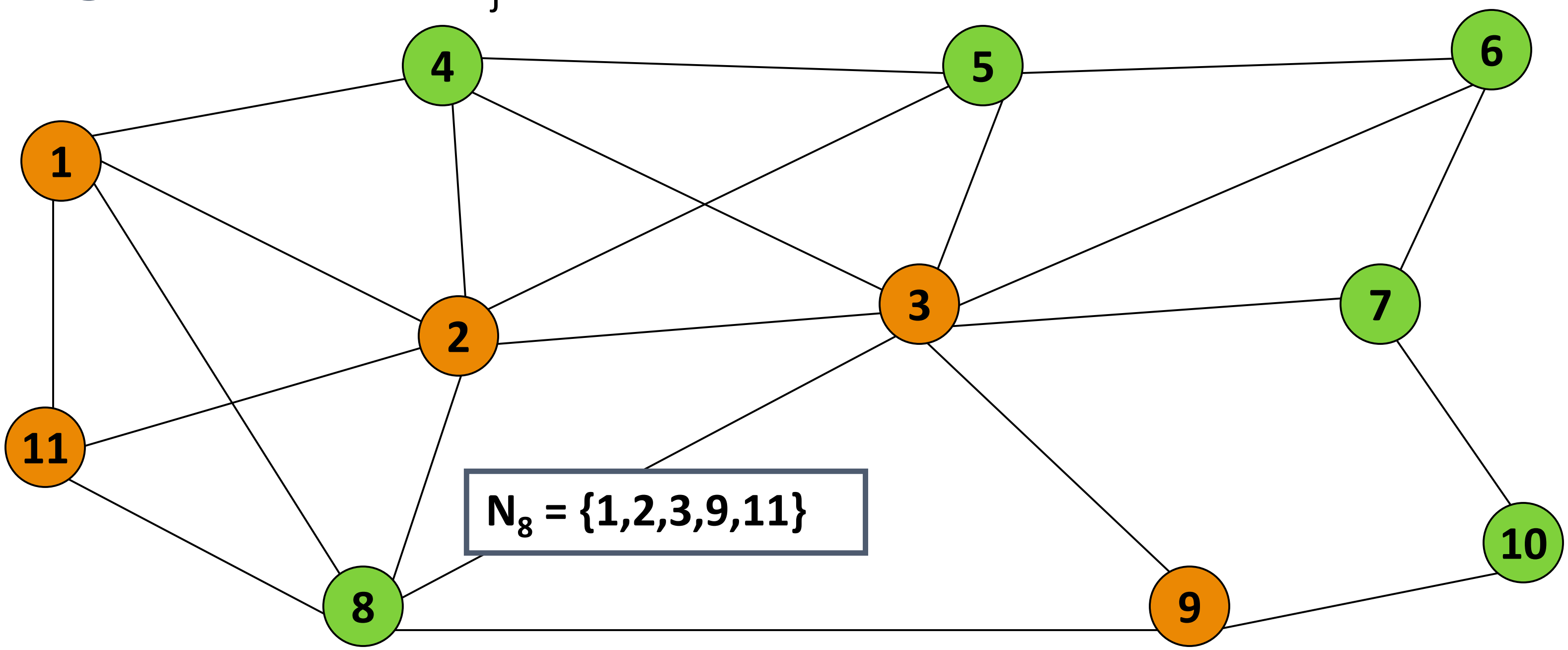
Maximal Independent Set Algorithm?

- For now, assume one process owns the entire graph. Ideas???????



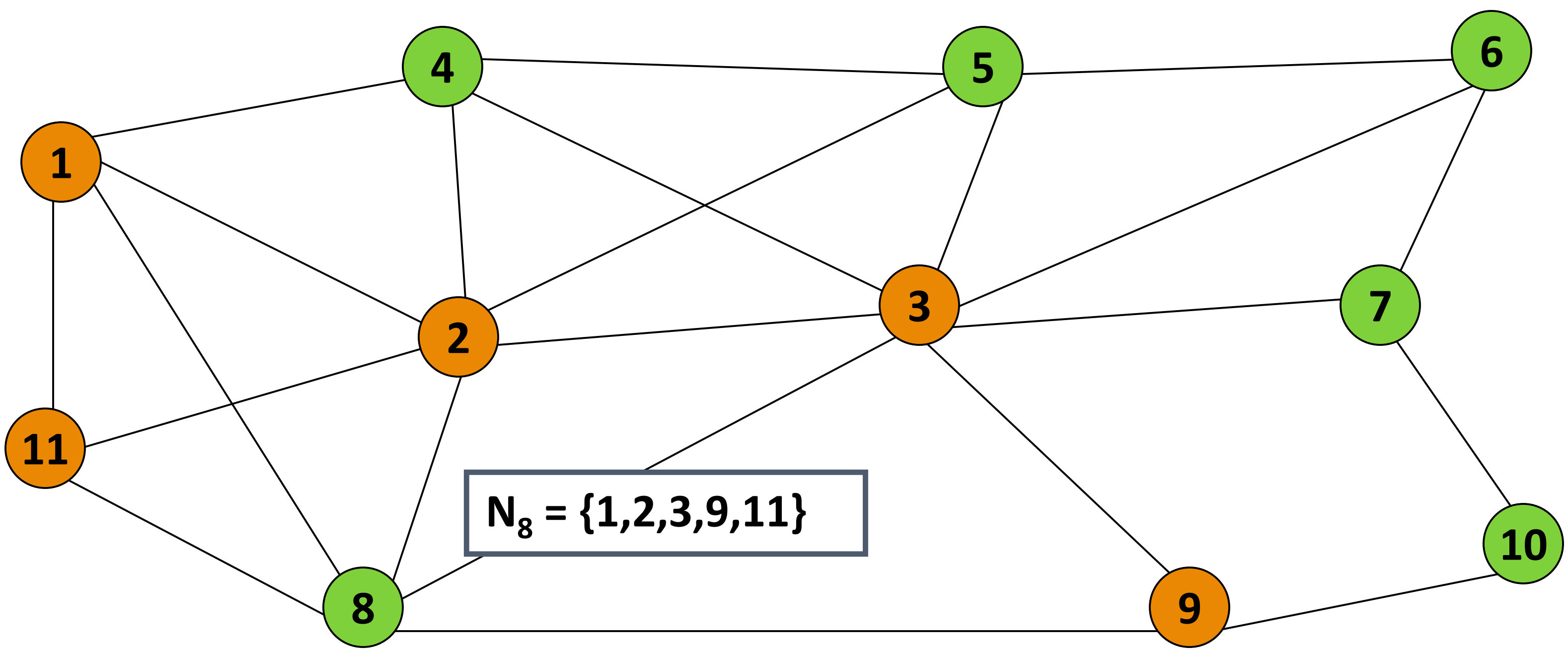
Maximal Independent Set Algorithm? Hint # 1

- The **neighborhood** N_j of a vertex v_j consists of all vertices connected to v_j by an edge.
- The number of vertcies in the neighborhood is called the **degree** of vertex v_j



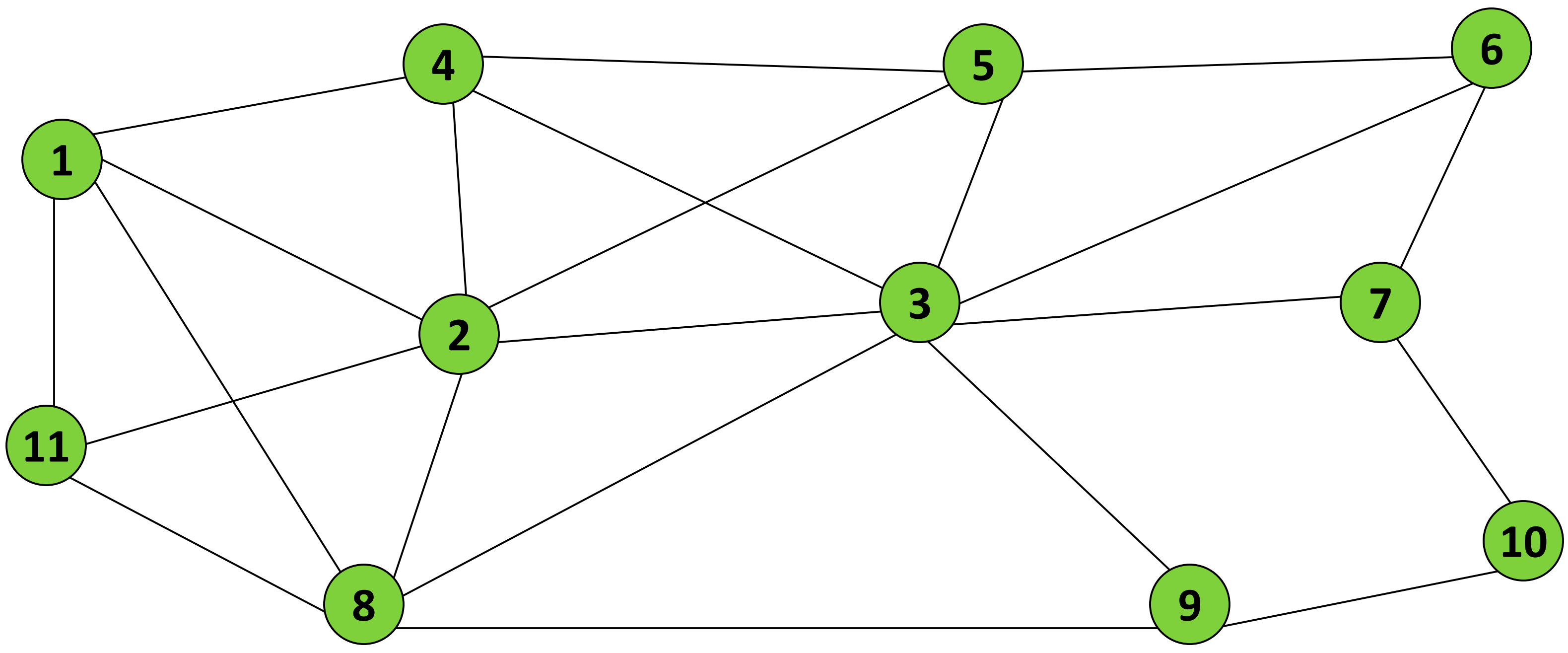
Maximal Independent Set Algorithm? Hint # 2

- If vertex v_j is in set S and S is a maximal independent set then all vertices in neighborhood N_j are not in S.



Maximal Independent Set Algorithm

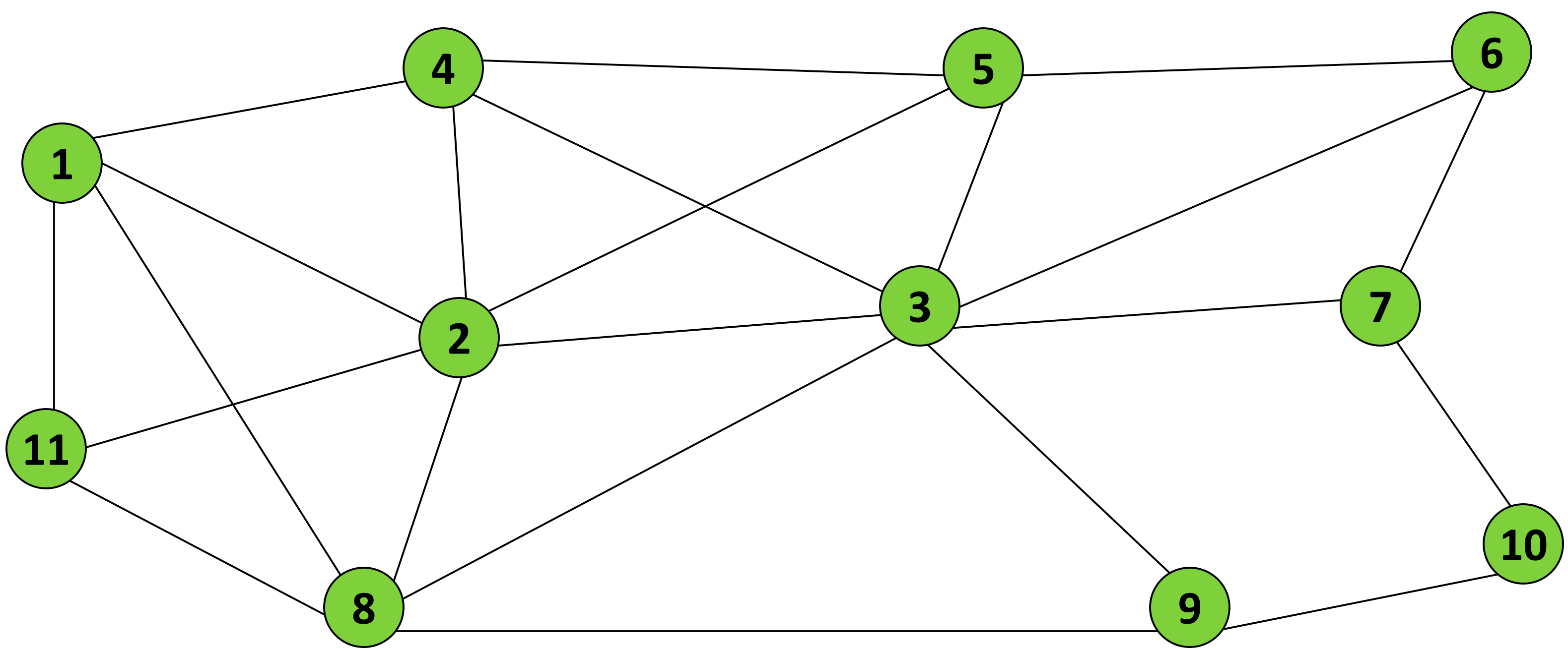
1. S=empty, U=V
2. While U not empty
   1. Pick v_j in U and add to S
   2. Remove v_j and its neighborhood N_j from U



Maximal Independent Set Algorithm

1. S=empty, U=V
2. While U not empty
    1. Pick v_j in U and add to S
    2. Remove v_j and its neighborhood N_j from U

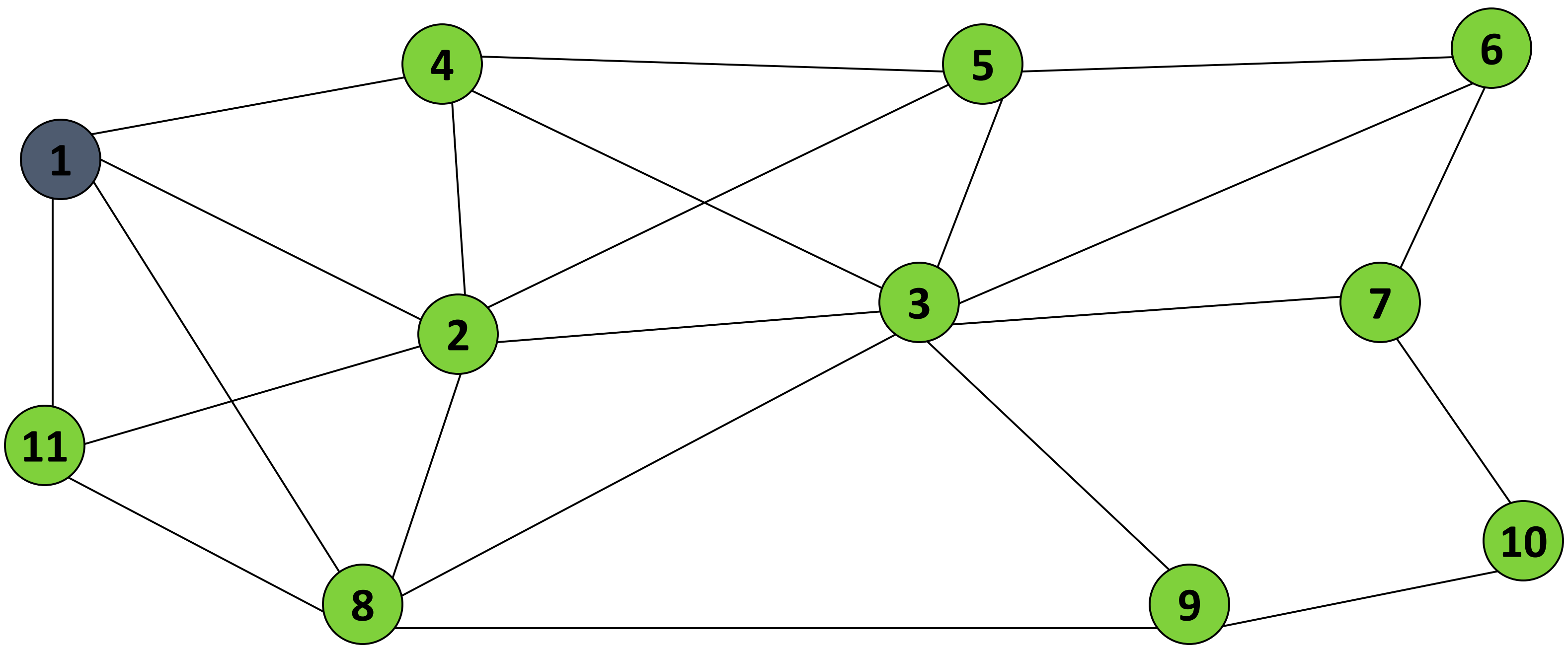
U={1,2,3,4,5,6,7,8,9,10,11}
S={ }



Maximal Independent Set Algorithm

1. **S=empty, U=V**
2. **While U not empty**
   1. **Pick v_j in U and add to S**
   2. **Remove v_j and its neighborhood N_j from U**

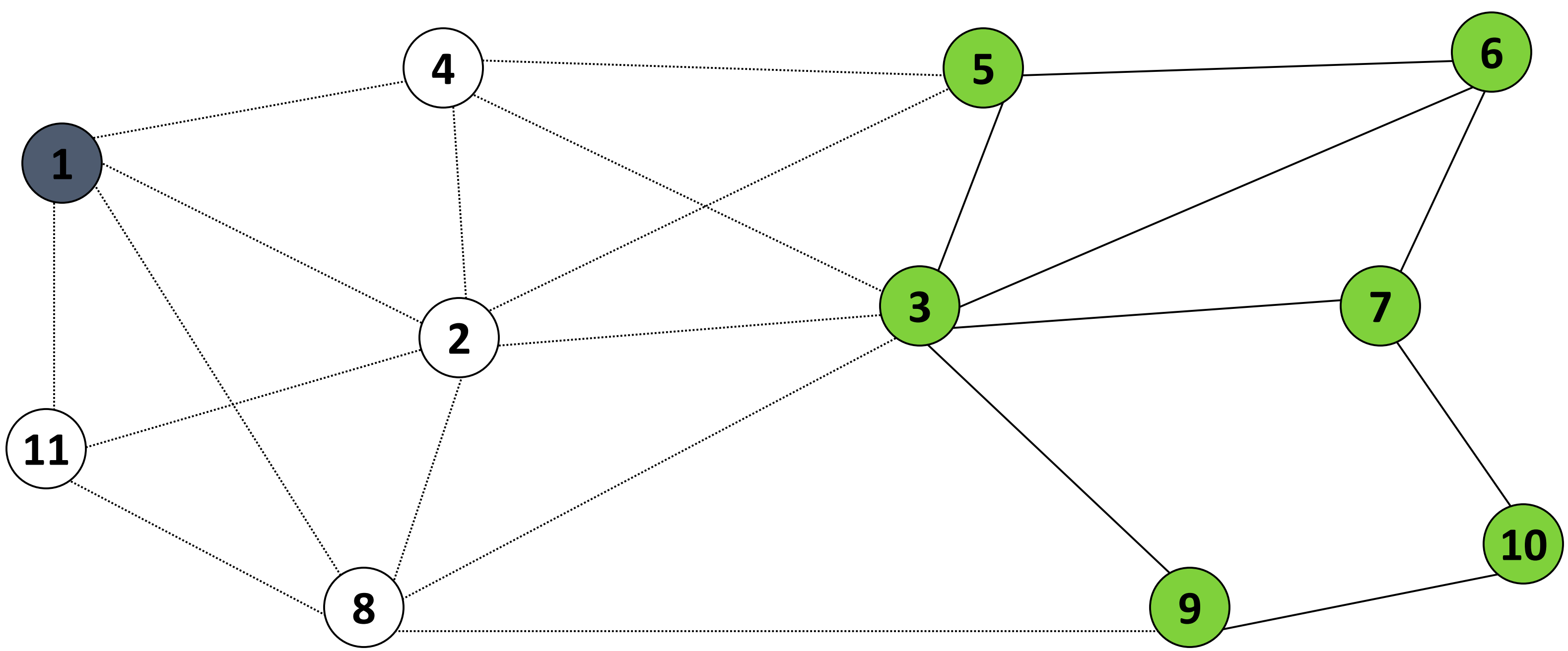
U={1,2,3,4,5,6,7,8,9,10,11}
S={1}



Maximal Independent Set Algorithm

1. **S=empty, U=V**
2. **While U not empty**
   1. **Pick v_j in U and add to S**
   2. **Remove v_j and its neighborhood N_j from U**

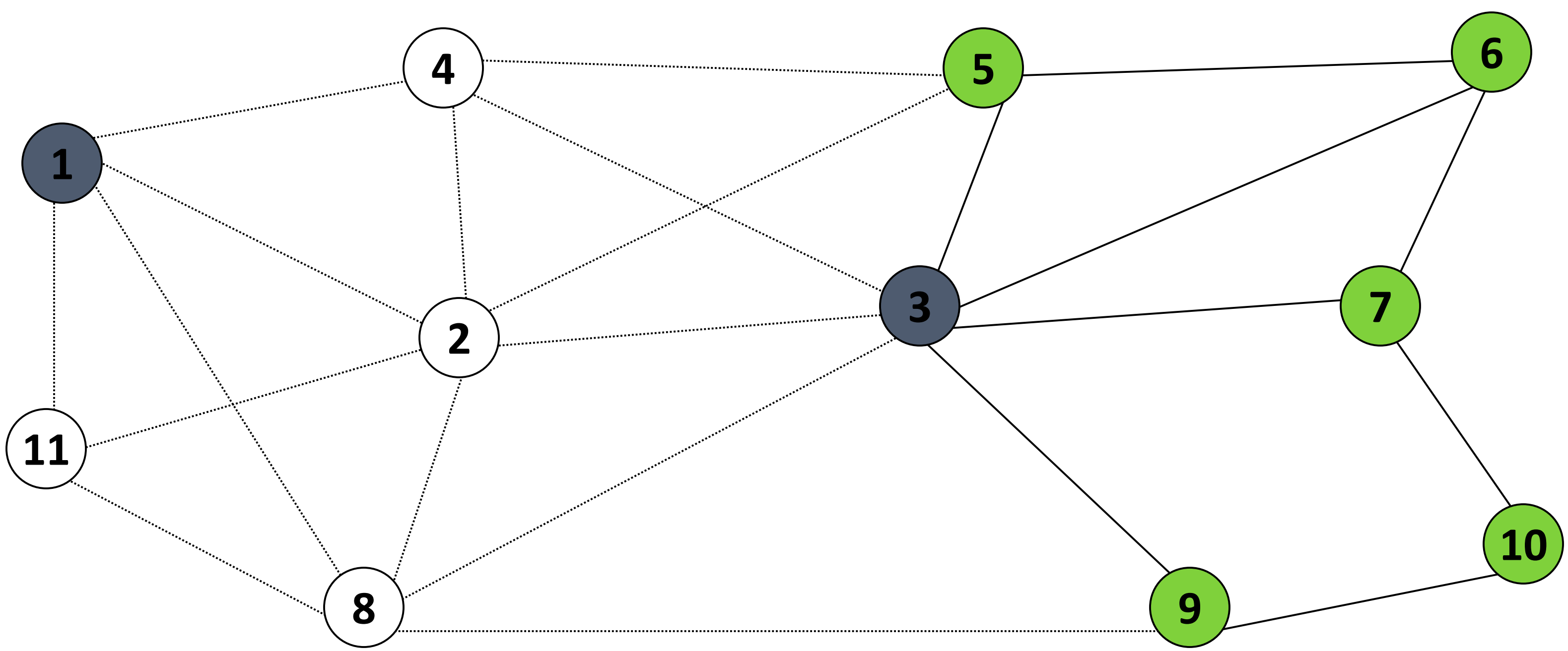
U={3,5,6,7,9,10}
S={1}



Maximal Independent Set Algorithm

1. S=empty, U=V
2. While U not empty
   1. Pick v_j in U and add to S
   2. Remove v_j and its neighborhood N_j from U

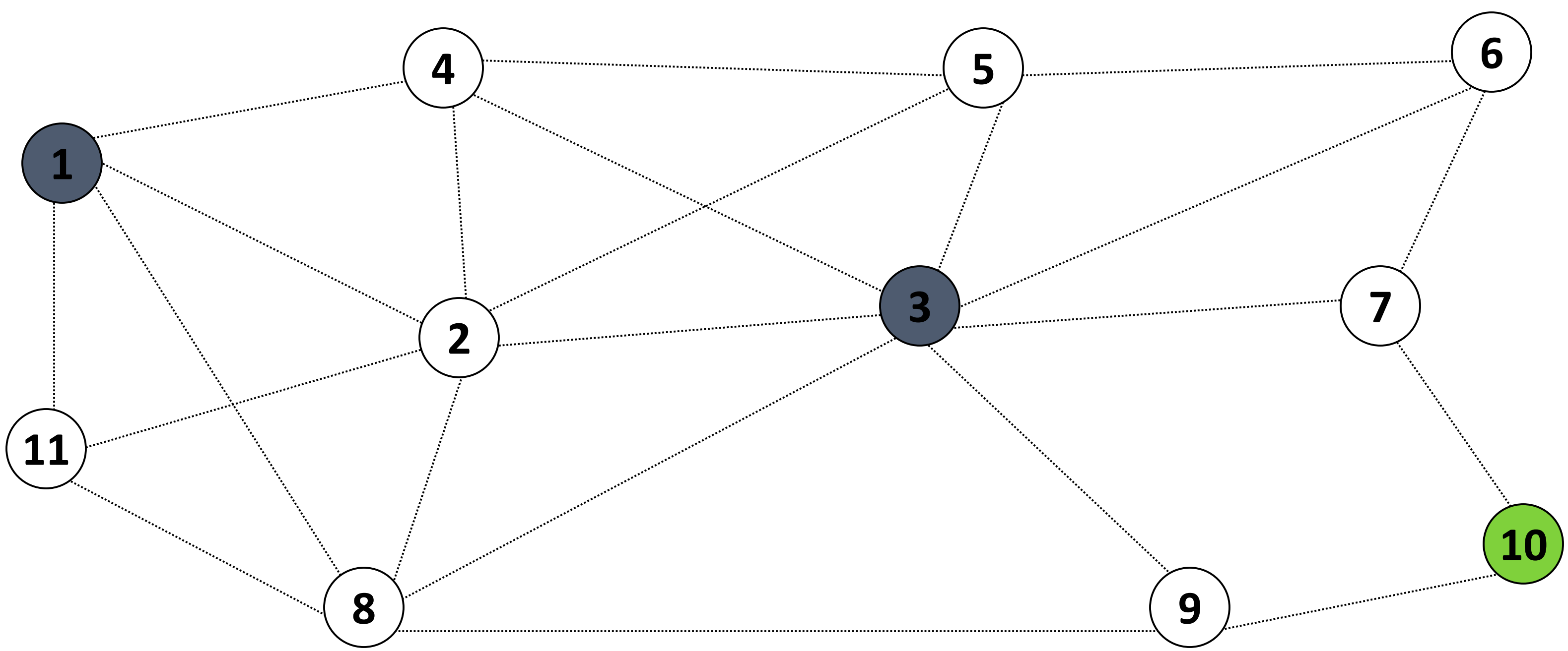
U={3,5,6,7,9,10}
S={1,3}



Maximal Independent Set Algorithm

1. S=empty, U=V
2. While U not empty
   1. Pick v_j in U and add to S
   2. Remove v_j and its neighborhood N_j from U

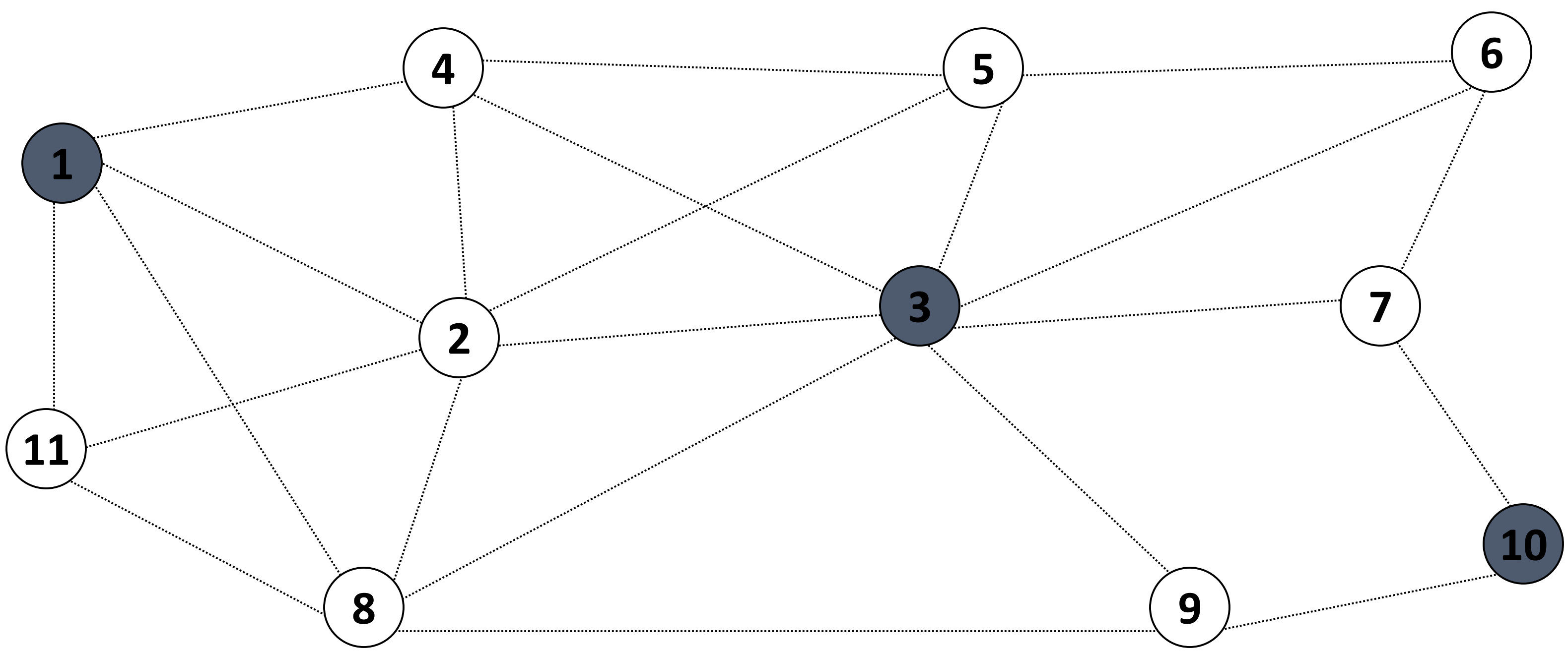
U={10}
S={1,3}



Maximal Independent Set Algorithm

1. **S=empty, U=V**
2. **While U not empty**
   1. **Pick v_j in U and add to S**
   2. **Remove v_j and its neighborhood N_j from U**

U={10}
S={1,3,10}

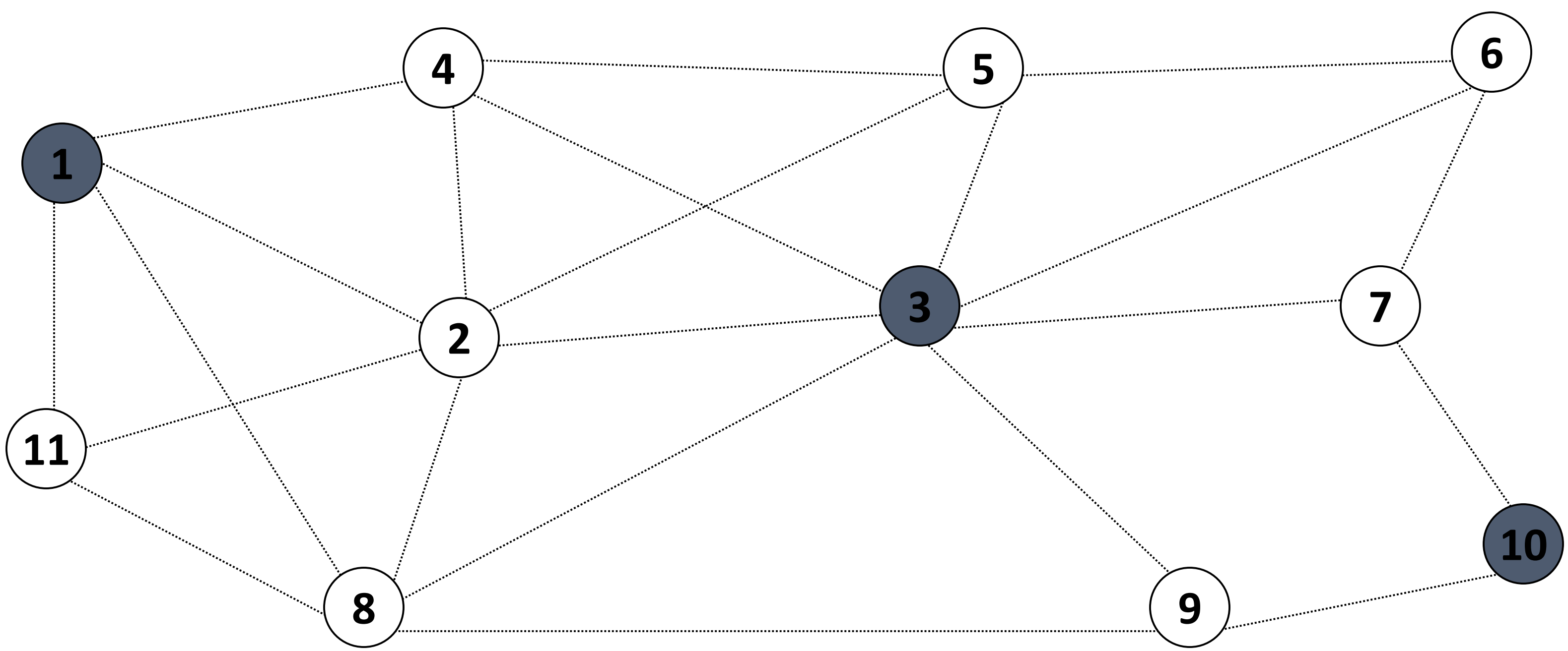


Maximal Independent Set Algorithm

1. S=empty, U=V
2. While U not empty
   1. Pick v_j in U and add to S
   2. Remove v_j and its neighborhood N_j from U

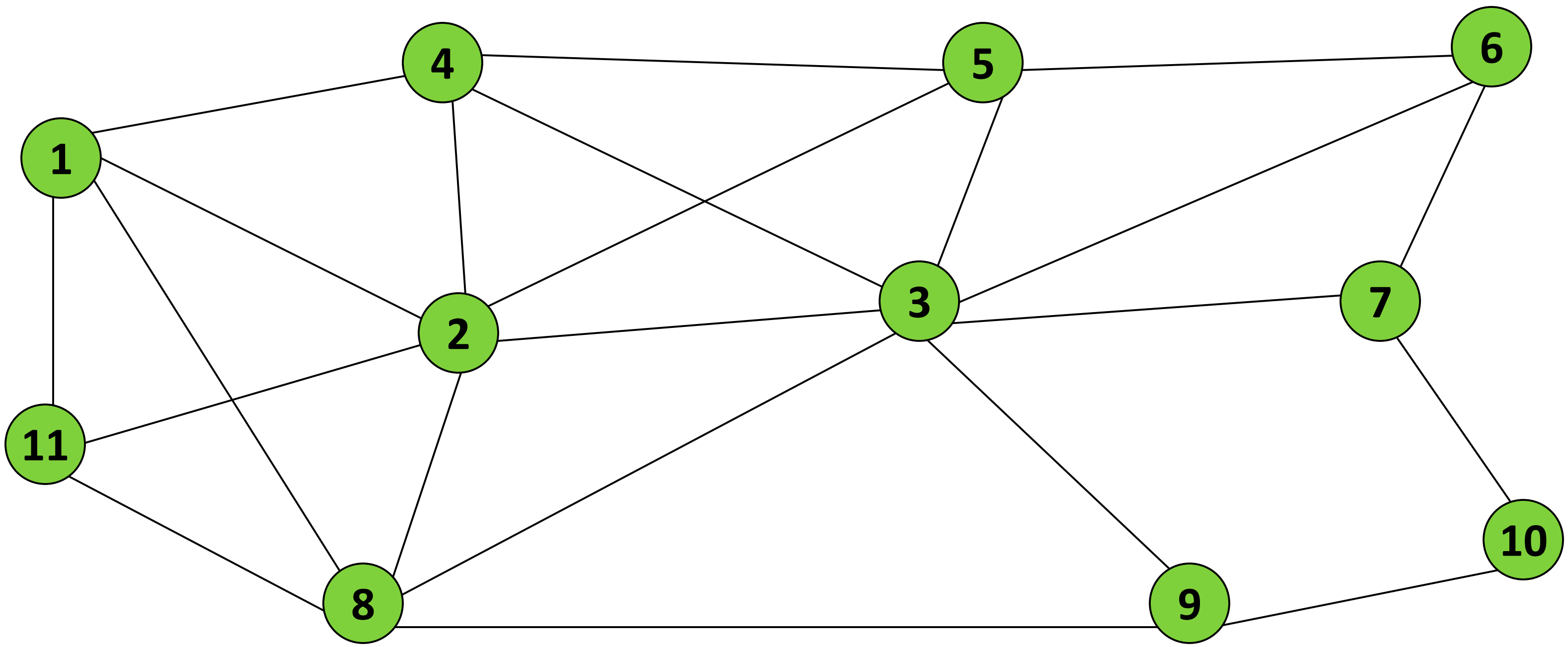
U={ }
S={1,3,10}

Done as U now empty



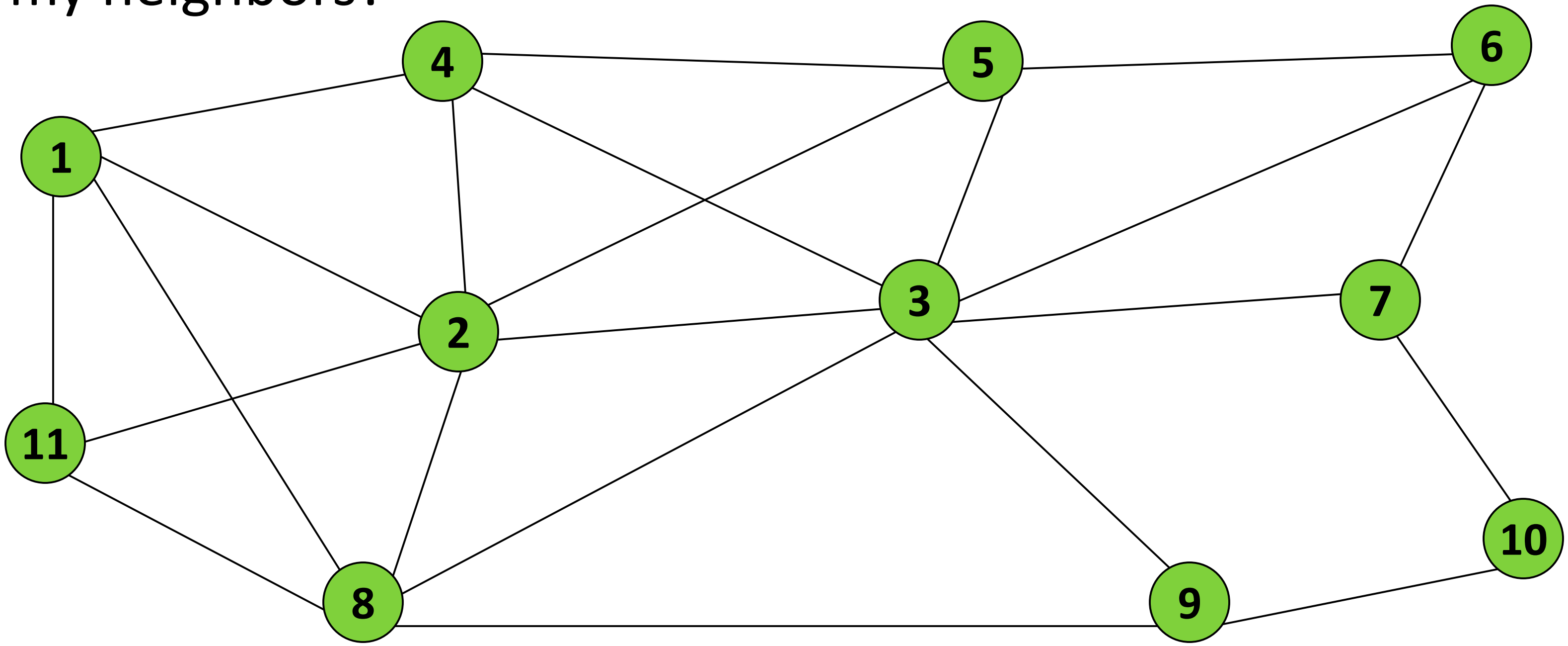
Maximal Independent Set Parallel Algorithm?

- Given a graph G=(V,E), find a maximal independent set S.
- Assume the number of processes is equal to n, the number of vertices.



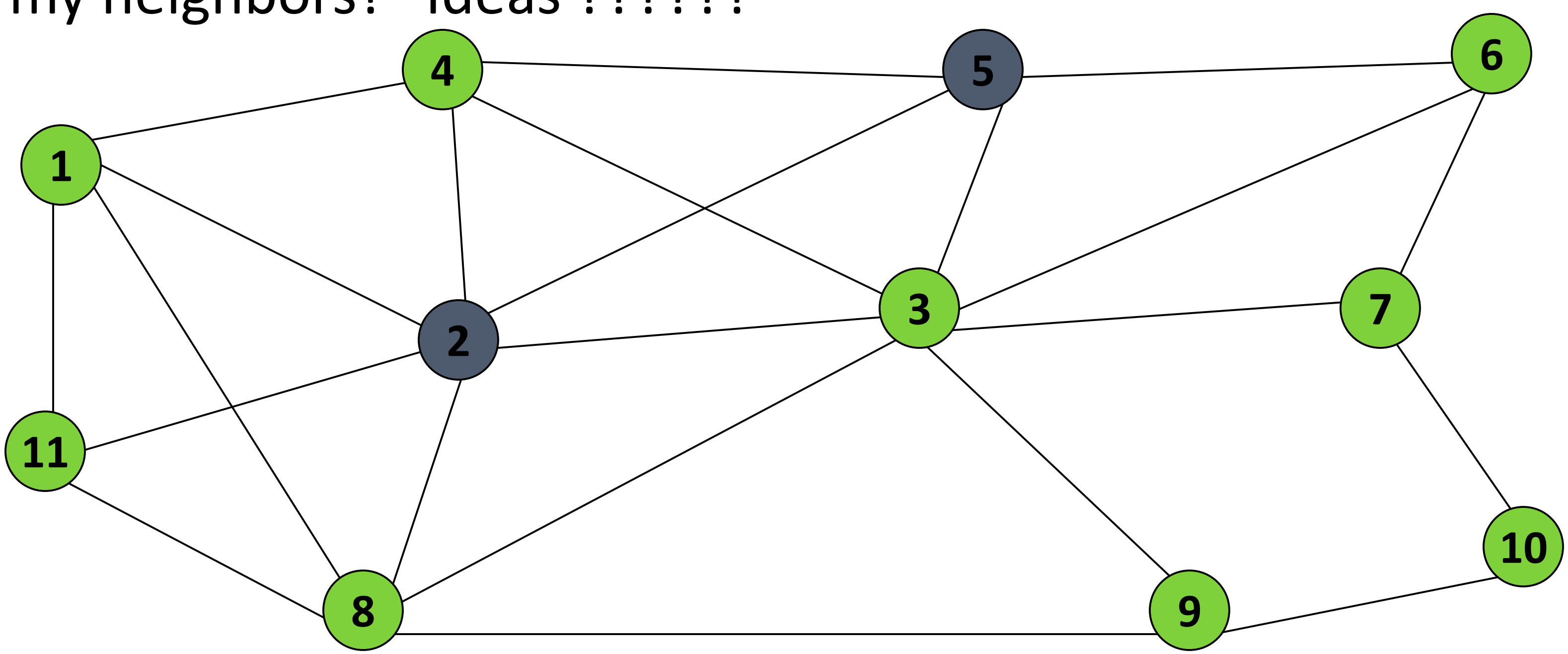
Maximal Independent Set Parallel Algorithm?

- Choosing if my vertex is in S or not is a candidate for the primitive task.
- Hurdle: how do I make sure my choice doesn't conflict with my neighbors?



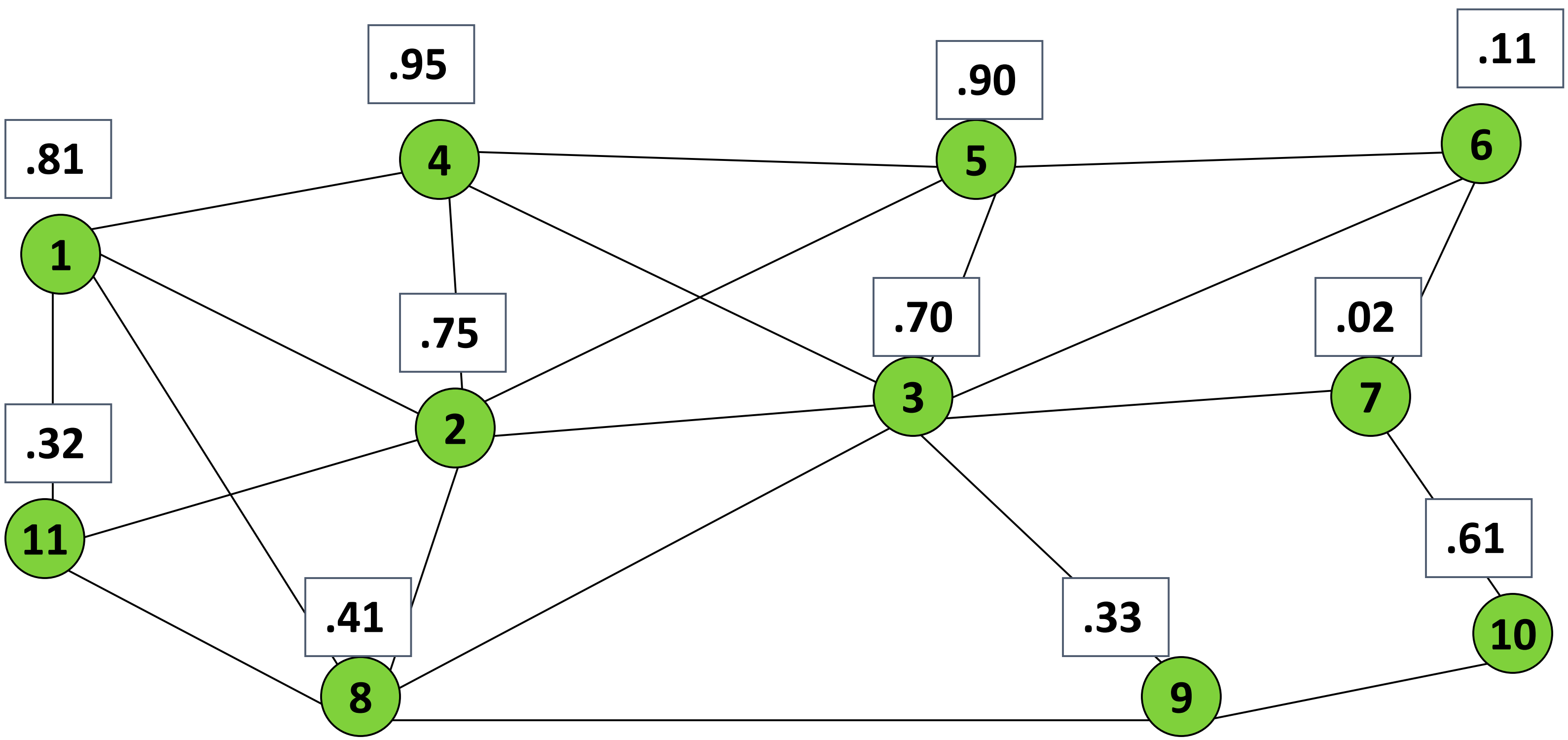
Maximal Independent Set Parallel Algorithm?

- Choosing if my vertex is in S or not is a candidate for the primitive task.
- Hurdle: how do I make sure my choice doesn't conflict with my neighbors? Ideas ??????



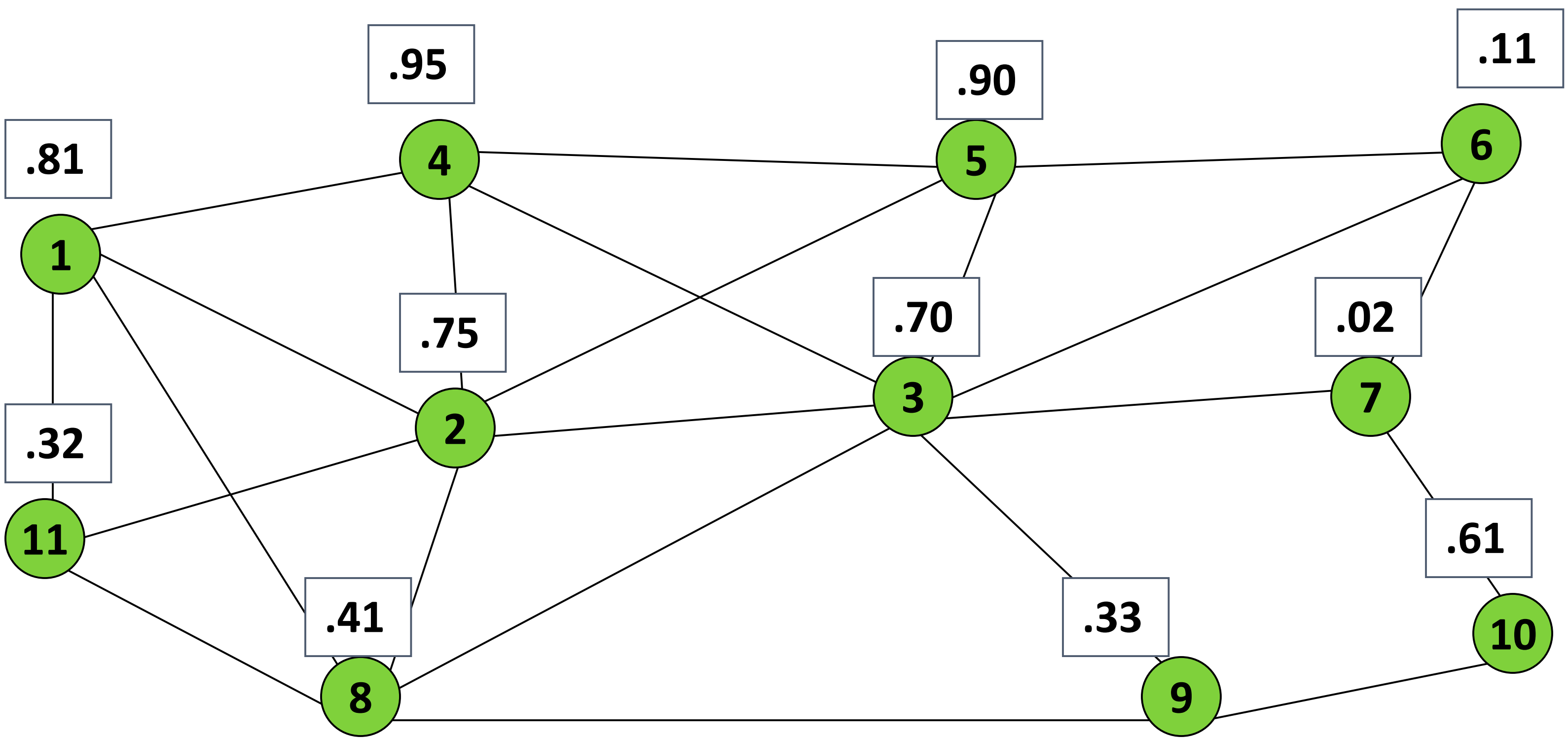
Maximal Independent Set Parallel Algorithm? Hint #1

- Assign a random number r_j in (0,1) to my vertex. Communicate this value to neighbors.



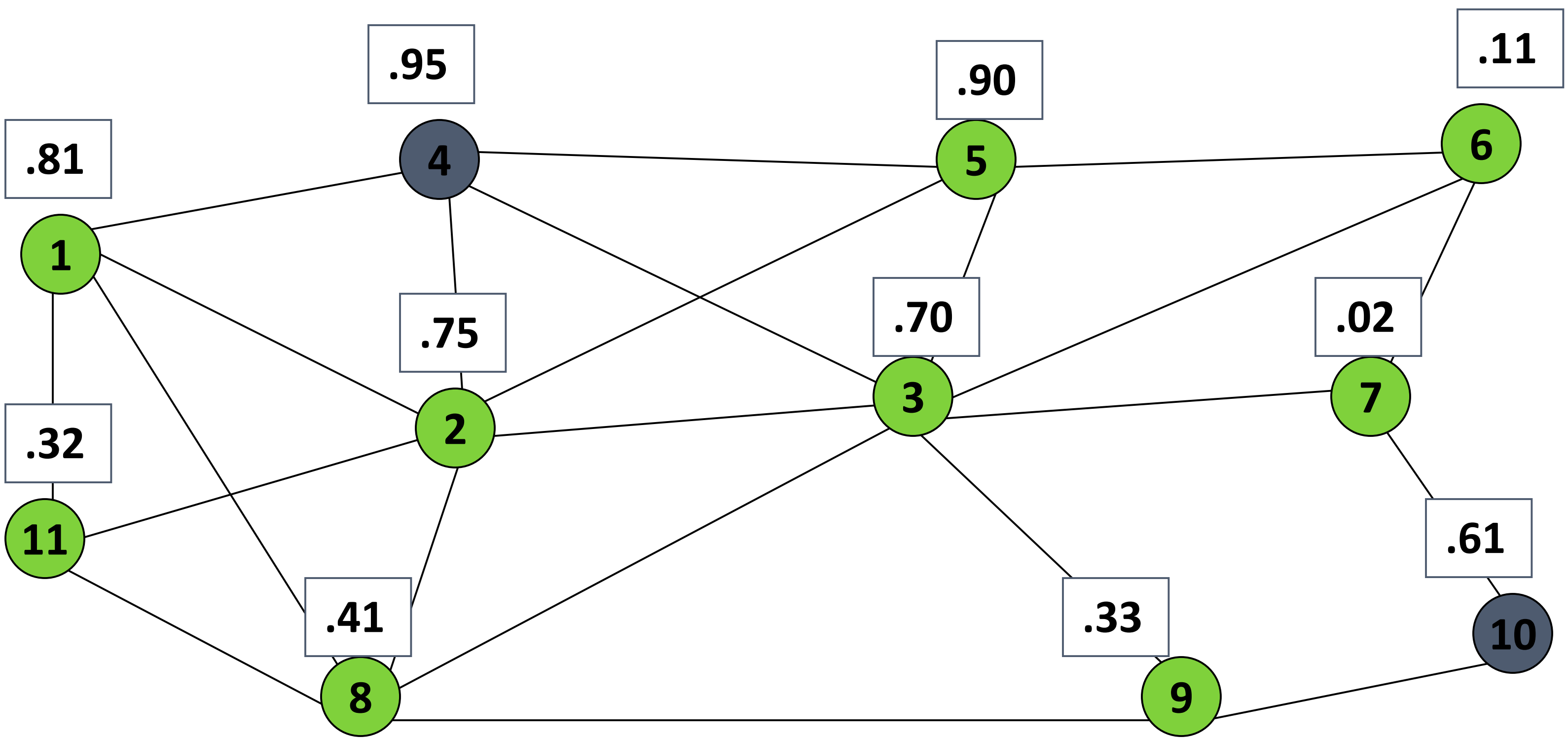
Maximal Independent Set Parallel Algorithm? Hint #2

- If my r is bigger than all my neighbor's r
  I'm in S.



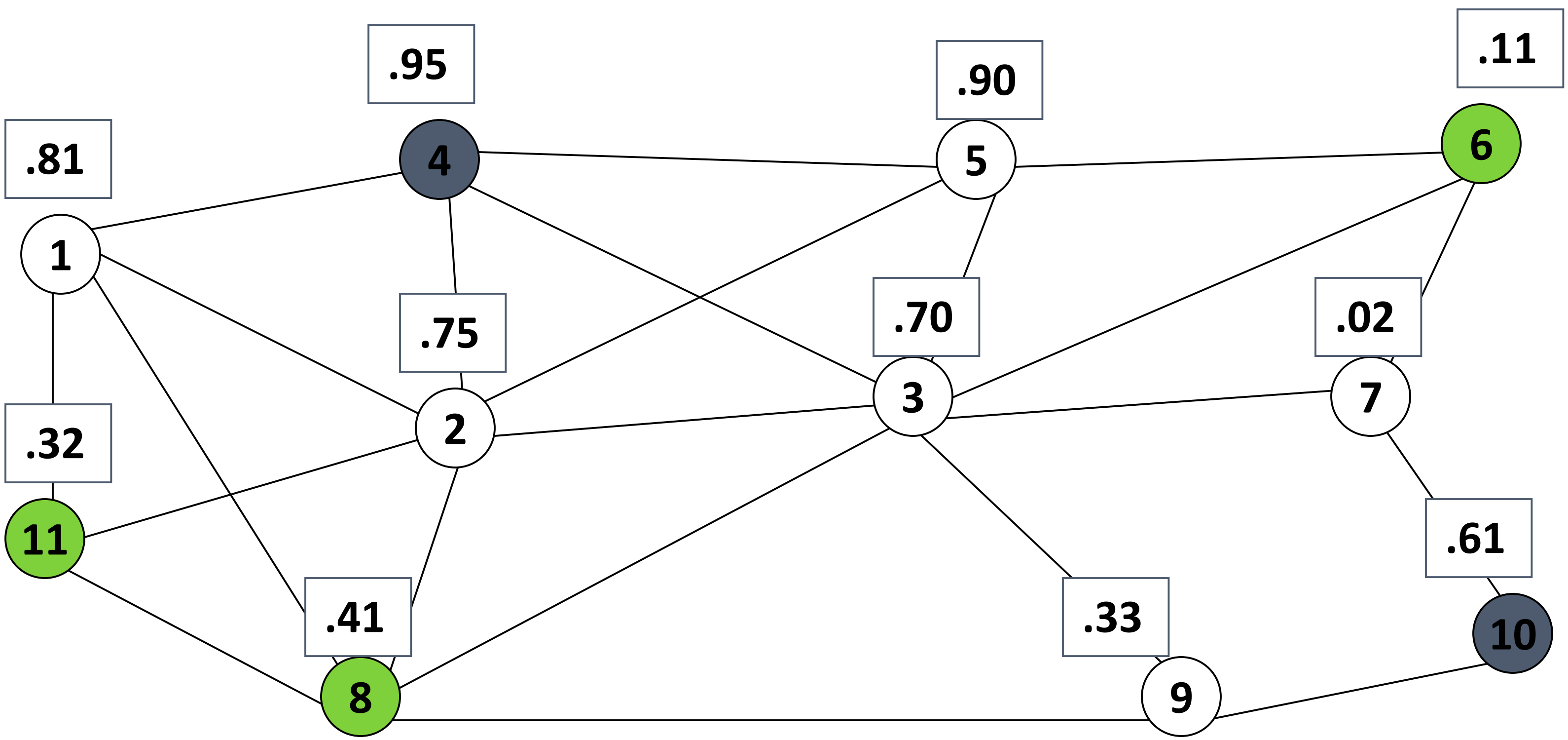
Maximal Independent Set Parallel Algorithm? Hint #2

- If my r is bigger than all my neighbor's r,
  I'm in S.



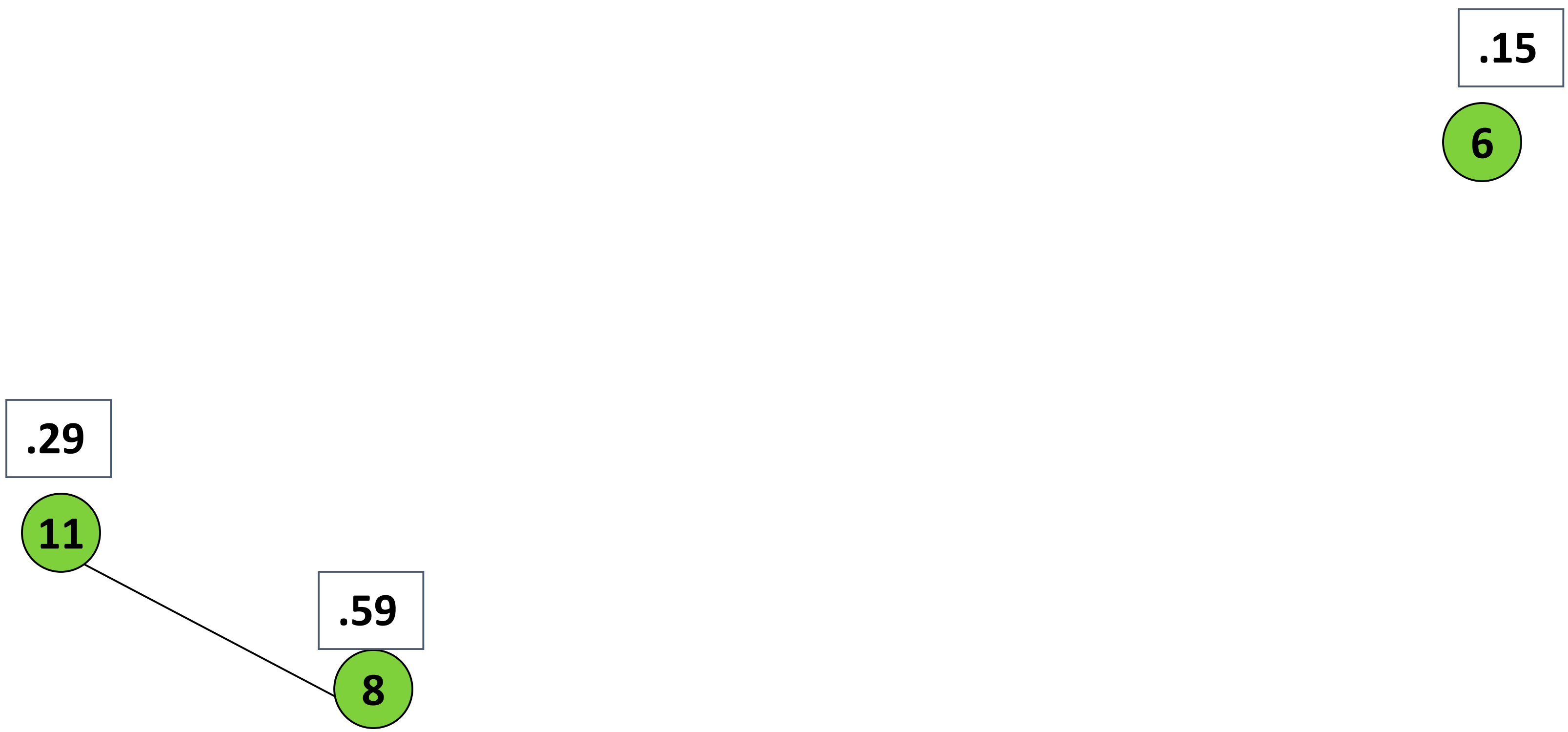
Maximal Independent Set Parallel Algorithm? Hint #3

- Remove vertices and edges for everyone in S and their neighbors (requires communication).



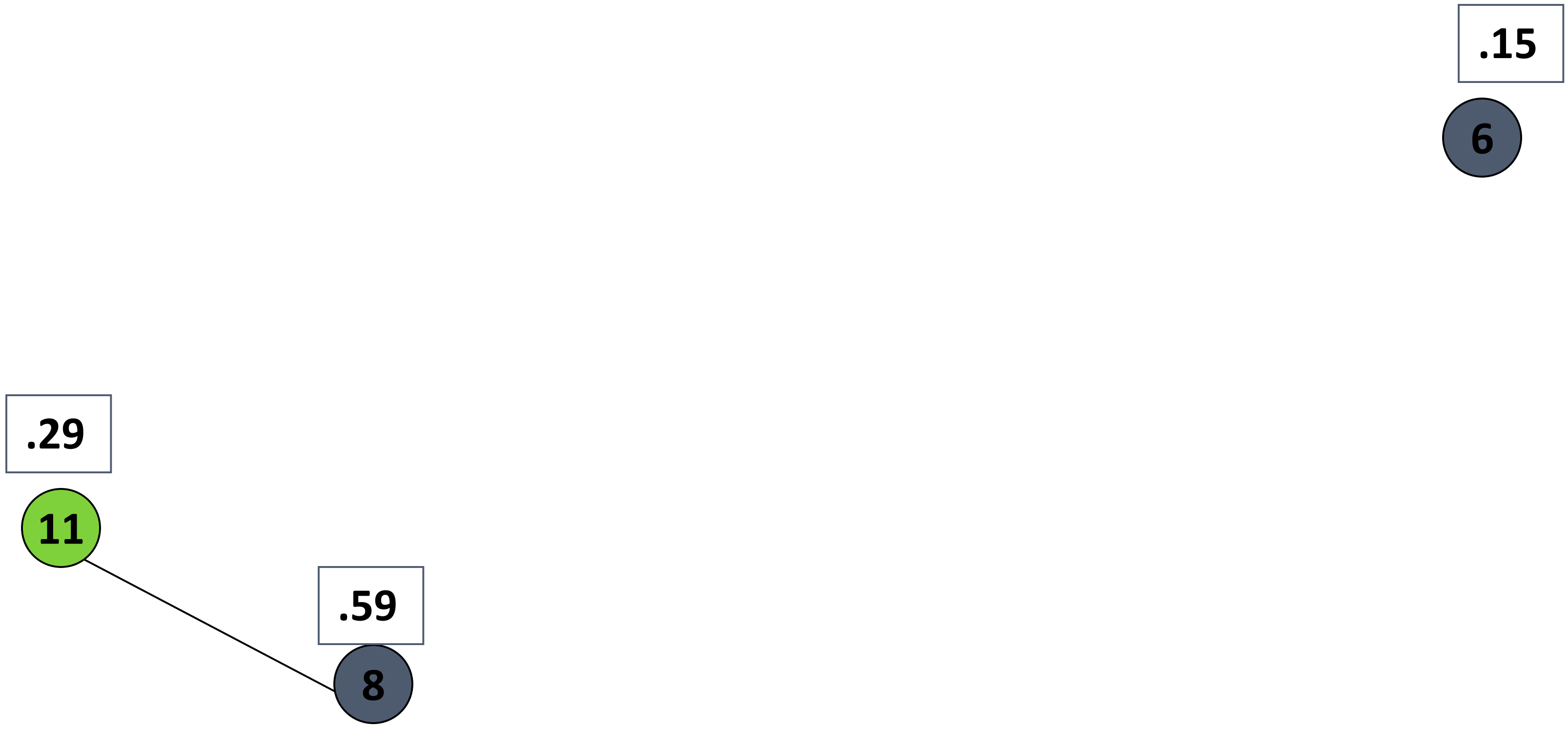
Maximal Independent Set Parallel Algorithm? Hint #3

- Start over with remaining graph.



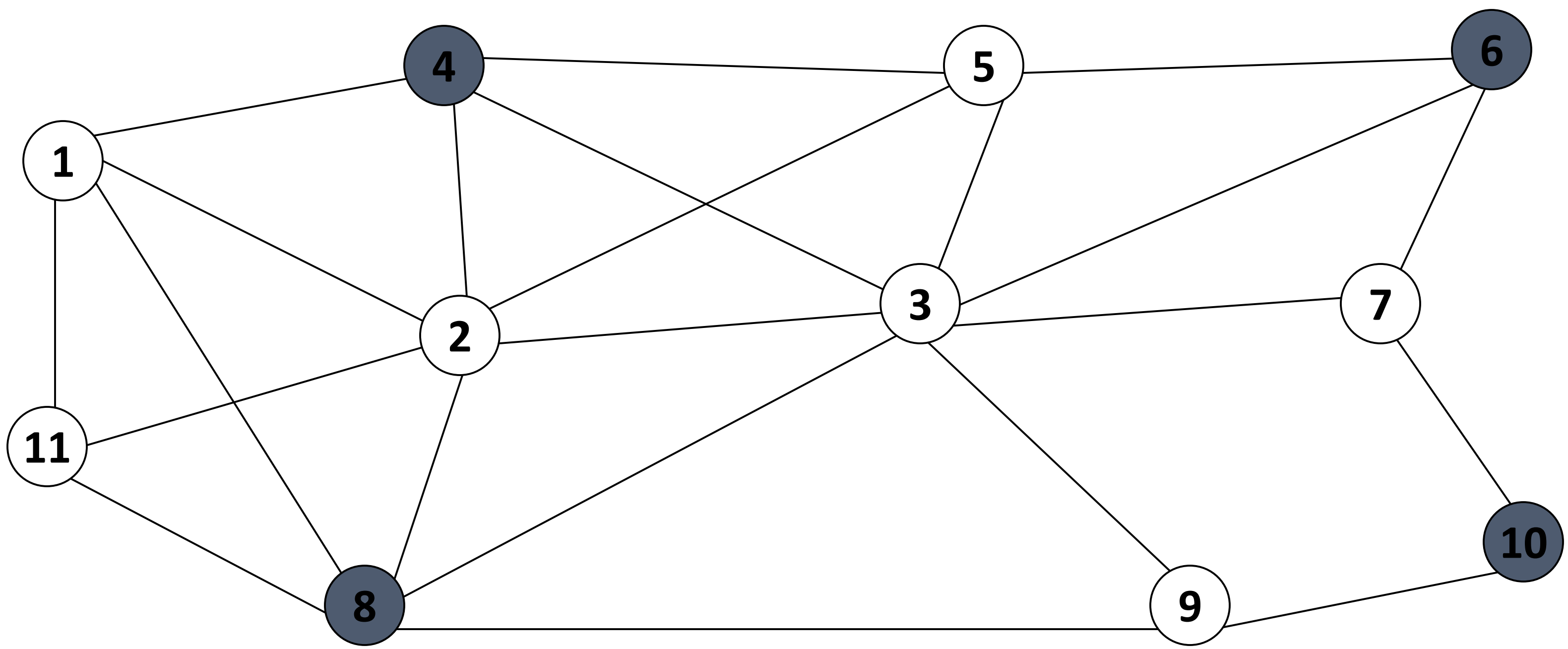
Maximal Independent Set Parallel Algorithm? Hint #3

- Start over with remaining graph.



Maximal Independent Set Algorithm

- No more undecided vertices.



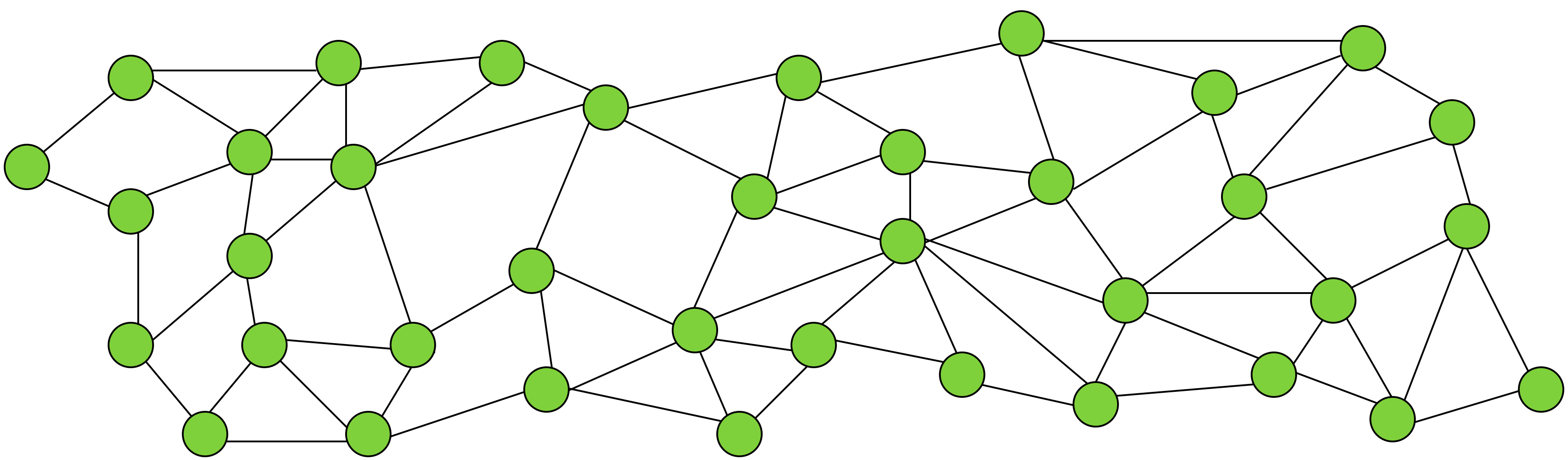
Maximal Independent Set Algorithm: Luby: SIAM J. Comput., 15(4):1036--1053

1. **S=empty, U=V**
2. **While U not empty**
   1. **Assign random r_i in (0,1) to each vertex**
   2. **If r_j is greater than neighbors add v_j to S***
   3. **Add S* to S**
   4. **Remove S* and its neighbors from U**

- Which MIS we get depends on the random numbers.
- Number of iterations of while loop depends on random numbers.

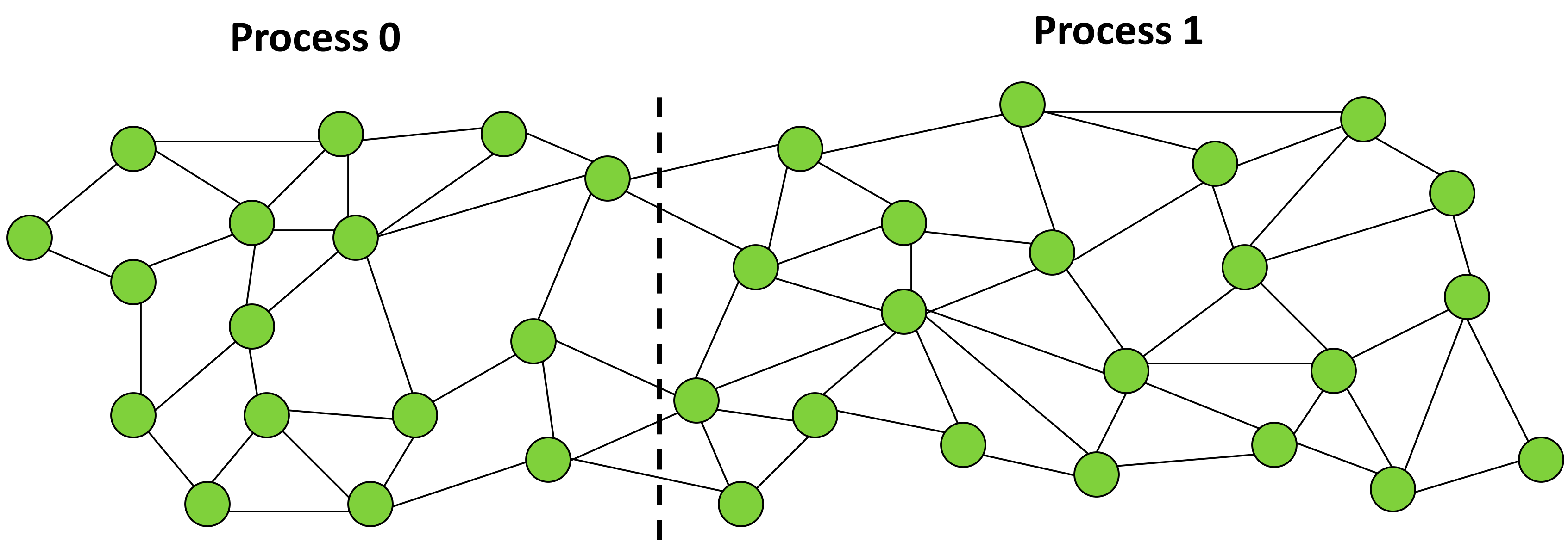
Maximal Independent Set Parallel Algorithm?

- Given a graph G=(V,E), find a maximal independent set S.
- Assume the number of processes is much smaller than n, the number of vertices.



Maximal Independent Set Parallel Algorithm?

- Given a graph G=(V,E), find a maximal independent set S.
- Assume the number of processes is much smaller than n, the number of vertices.
- The graph is distributed. Ideas?



Graph Coloring

- Develop Parallel Code.
- Run on Blueshark.
- Investigate speed-ups and scalability.

Possibilities

- Numerical PDEs
- Numerical Linear Algebra
  – Sparse Matrix times Vector
- Monte Carlo Methods
  – Neutron Transport
- Sorting
- Others: searching, dense linear algebra, maximal independent sets, graph coloring …
- Any topic from texts that we haven't covered
  – Pacheco: n-Body Solvers and Tree Search
  – Quinn: lots – sorting, searching, FFT, Matrix Algebra
- Other styles of parallelism on other machines: like GPU. "Proof of concept" recommended before proposal.
- Same originality "Rules" as assignments: the code your team turns in must be written by your team.