Quiz due Monday 6/2

Travel the world!
See the piazza post @630 with combined week 10 office hours.

First batch of sample final questions are posted, solutions and other problems to follow

Week 10 Take Home quiz due Wednesday 6/4.

Wednesday (6/4) 6-7:30pm (EBU3B 4122)
Wednesday (6/4) 8-10pm (Homeplate)
Thursday (6/5) 8-10pm (Homeplate)
Friday (6/6) 12-1pm (EBU3B 4122)
Friday (6/6) 2-5pm at The Loft (with tutors / TAs)

Final Exam Study Session 1: Fri. 6/6 at 6-8pm (CENTER 115)
Saturday (6/7) 5-6pm (EBU3B 4122)

Final Exam Study Session 2: Sat. 6/7 at 6-8pm (CENTER 119)
Special families of graphs - Trees

Any two conditions imply the third condition

- $T$ is a tree on $N$ vertices
- $T$ has $N-1$ edges
- $T$ is connected

This graph has 5 vertices and 4 edges, but it is not a tree.
Trees - SAGE code used in class

We looked at several random graphs with 10 vertices and 9 edges. If the graph is connected, then this graph will be a tree.

https://sagecell.sagemath.org

T=graphs.RandomGNM(10,9)
T.plot()

Not a tree, since the graph is disconnected
Trees - SAGE code used in class

We looked at several random graphs with 10 vertices and 9 edges. If the graph is connected, then this graph will be a tree.

https://sagecell.sagemath.org

T = graphs.RandomGNM(10, 9)
T.plot()

Not a tree, since the graph is disconnected
We looked at several random graphs with 10 vertices and 9 edges. If the graph is connected, then this graph will be a tree.

https://sagecell.sagemath.org

T = graphs.RandomGNM(10, 9)
T.plot()
Trees - SAGE code used in class

You can also ask SAGE to generate random trees!

https://sagecell.sagemath.org

T=graphs.RandomTree(10)
T.plot()

This is a tree on 10 vertices!
You can also ask SAGE to generate random trees!

https://sagecell.sagemath.org

T=graphs.RandomTree(20)
T.plot()
Trees - SAGE code used in class

You can also ask SAGE to generate random trees!

https://sagecell.sagemath.org

T=graphs.RandomTree(30)
T.plot()

This is a tree on 30 vertices!
If you can partition a graph into two sets $X$ and $Y$, with no edges inside the sets $X$ or $Y$, then the graph is called bipartite.
If you can partition a graph into two sets $X$ and $Y$, with no edges inside the sets $X$ or $Y$, then the graph is called bipartite.

Equivalently all edges have one end in $X$ and the other end in $Y$. 
If you can partition a graph into two sets $X$ and $Y$, with no edges inside the sets $X$ or $Y$, then the graph is called bipartite. Equivalently all edges have one end in $X$ and the other end in $Y$. A graph is bipartite if and only if it contains no odd cycles.
Complete Graphs - SAGE code used in class

We asked SAGE to plot several complete graphs on $N$ vertices, denoted as $K_N$, these graphs all have $N$ vertices and all possible \( \binom{N}{2} \) edges.

```python
K=graphs.CompleteGraph(15)
K.plot()
```
Complete Graphs - SAGE code used in class

\( K_{16} \), the complete graph on \( N = 16 \) vertices and all possible \( \binom{16}{2} \) edges.

\[
K = \text{graphs.CompleteGraph}(16)
\]
\[
K.plot()
\]
Complete Graphs - SAGE code used in class

\[ K_{17}, \text{ the complete graph on } N = 17 \text{ vertices and all possible } \binom{17}{2} \text{ edges.} \]

\[
K = \text{graphs.CompleteGraph}(17) \\
K.\text{plot}()
\]
$K_{18}$, the complete graph on $N = 18$ vertices and all possible $\binom{18}{2}$ edges.

K=graphs.CompleteGraph(18)
K.plot()}
$K_{19}$, the complete graph on $N = 19$ vertices and all possible $\binom{19}{2}$ edges.

K=graphs.CompleteGraph(19)
K.plot()
$K_{20}$, the complete graph on $N = 20$ vertices and all possible $\binom{20}{2}$ edges.

```
K=graphs.CompleteGraph(20)
K.plot()
```
$K_{21}$, the complete graph on $N = 21$ vertices and all possible $\binom{21}{2}$ edges.

```python
K=graphs.CompleteGraph(21)
K.plot()
```
Complete Graphs - SAGE code used in class

$K_{22}$, the complete graph on $N = 22$ vertices and all possible $\binom{22}{2}$ edges.

```python
K=graphs.CompleteGraph(22)
K.plot()
```
$K_{23}$, the complete graph on $N = 23$ vertices and all possible $\binom{23}{2}$ edges.

```
K=graphs.CompleteGraph(23)
K.plot()
```
Complete Graphs - SAGE code used in class

$K_{24}$, the complete graph on $N = 24$ vertices and all possible $\binom{24}{2}$ edges.

```
K=graphs.CompleteGraph(24)
K.plot()
```
$K_{25}$, the complete graph on $N = 25$ vertices and all possible $\binom{25}{2}$ edges.

```python
K = graphs.CompleteGraph(25)
K.plot()
```
$K_{26}$, the complete graph on $N = 26$ vertices and all possible $\binom{26}{2}$ edges.

```python
K = graphs.CompleteGraph(26)
K.plot()
```
**Complete Graphs - SAGE code used in class**

\( K_{27} \), the complete graph on \( N = 27 \) vertices and all possible \( \binom{27}{2} \) edges.

```python
K=graphs.CompleteGraph(27)
K.plot()
```
$K_{28}$, the complete graph on $N = 28$ vertices and all possible $\binom{28}{2}$ edges.

K=graphs.CompleteGraph(28)
K.plot()
$K_{29}$, the complete graph on $N = 29$ vertices and all possible $\binom{29}{2}$ edges.

```python
K=graphs.CompleteGraph(29)
K.plot()
```
$K_{30}$, the complete graph on $N = 30$ vertices and all possible $\binom{30}{2}$ edges.

```python
K = graphs.CompleteGraph(30)
K.plot()
```
$K_{31}$, the complete graph on $N = 31$ vertices and all possible $\binom{31}{2}$ edges.

```python
K=graphs.CompleteGraph(31)
K.plot()
```
$K_{32}$, the complete graph on $N = 32$ vertices and all possible $\binom{32}{2}$ edges.

\begin{verbatim}
K=graphs.CompleteGraph(32)
K.plot()
\end{verbatim}
Recall that a proper coloring is a function from the vertices of the graph $V(G)$ to a set of colors $\{c_1, c_2, c_3, \ldots, c_k\}$ so that no two adjacent vertices are assigned the same color. The minimum value for which there exists a proper $k$-coloring is called the chromatic number, denoted as $\chi(G)$.

- $\chi(C_{2a+1}) = 3$, for all integers $a \geq 1$ (Odd cycles require three colors).
- If $G$ is bipartite, then $\chi(G) = 2$
- $\chi(C_{2a}) = 2$, for all integers $a \geq 2$ (even cycles are bipartite)
- Let $T$ be a tree. $\chi(T) = 2$. (Trees are bipartite)
- Let $K_N$ be the complete graph on $N$ vertices, then $\chi(K_N) = N$
Coloring Complete Graphs - SAGE code used in class

\( K_7 \), the complete graph on \( N = 7 \) vertices and all possible \( \binom{7}{2} \) edges.

\[
K=\text{graphs.CompleteGraph}(7) \\
K.\text{plot}() \\
K.\text{chromatic\_number}()
\]

\( \chi(K_7) = 7 \) since every vertex is adjacent to every other vertex, each vertex must receive a different color.
Week 10 Quiz - Due Wed 6/6

For Q1-Q4, properly color the vertices of the graph using the minimum number of colors.

Q1. $P_2$

Q2. $\mu(P_2) = C_5$

Q3. $\mu(\mu(P_2)) = \mu(C_5) =$ The Grötzsch graph

Q4. A random tree on 10 vertices.

E.C. $\chi[\mu(\mu(\mu(P_2))))] = \chi[\mu(\text{the Grötzsch graph})] =$