Problem Set 2

Math 565: Combinatorics and Graph Theory

Prof. Thomas Fun Yau Lam

Due Time: Sep 23 at 2:30p.m.

Author: **Yanzhi Li**

Collaborators: Lingwei Zhang

1 Problem 1

Problem 1.1. Prove that $N(4,4;2) = 18$.

Solution:

Proof. To see it, we first want to see if for a K_{18}, we will color it with blue and red, in particular two different colors, then we see if we can obtain either a red K_4 or a blue K_4 as subgraph. Then we want to see that for a two-colored K_{17}, there exists a concrete counterexample that the graph will not induce either a red K_4 or a blue K_4 as subgraph.

Note that previously in the lecture, we have seen that:

$$N(3,4;2) = 9$$

And we setup our $K_{18} = (V_1, E_1)$ and $K_{17} = (V_2, E_2)$ by coloring their edges in either blue or red color.

- $\mathbf{K_{18}}$ **works**: We arbitrarily pick a vertex $A \in V_1$, by definition of a complete graph it will adjacent to 17 other vertices, thus will induce 17 edges. By **Pigeonhole Principle**, in this 17 edges, either:

$$\#(\text{red edges}) \geq 9$$
$$\#(\text{blue edges}) \geq 9$$

  So we then look into this two cases:

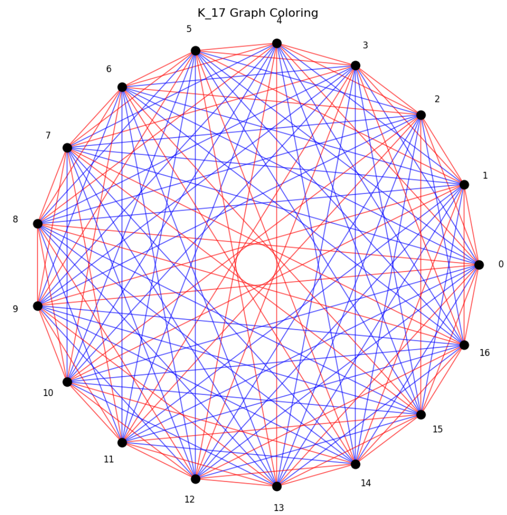
  1. **#(red edges)**$\geq$ **9**: We then denote all the adjacent vertices with red edges connected to A along with A to be the vertex set P, in particular $|P| = 10 > N(3,4;2) = 9$. Then by **Ramsey's Theorem**, in this subgraph induced by vertex set P, either there is a red K_3 or there is a blue K_4. And this K_3 along with A will form another red K_4 since the vertices in this K_3 all connect with A with red edges. So for the whole K_{18}, there exists either a red K_4 or a blue K_4 as subgraph.
  2. **#(blue edges)** $\geq$ **9**: The argument is the same as what we prove for the case **#(red edges)**$\geq$ **9** based on the fact that $N(3,4;2) = N(4,3;2) = 9$.

  So we see that K_{18} indeed works, in particular we have proven:

$$N(4,4;2) \leq 18$$

- $\mathbf{K_{17}}$ **doesn't work**: We shall construct a counterexample that shows in K_{17} there exists a special coloring that contains no red K_4 and blue K_4 as subgraph. In particular this leads to $N(4,4;2) \neq 17$. We proceed our construction as follows:

  Given a K_{17}, we label each vertices as $0, 1, 2, \ldots, 16$. Given $\forall i, j \in \{0, 1, \ldots, 16\}$, if $\min\{|i-j|, 17 - |i-j|\} \in \{1, 2, 4, 8\}$, we will color such edges in **red**, and if $\min\{|i-j|, 17 - |i-j|\} \in \{3, 5, 6, 7\}$, we will color such edges in **blue**. And one can verify that such construction contains no blue K_4 and red K_4. A picture of such construction can be find in following picture, and is enumerated to be correct by code simulation, the python source code of the code simulation can be found at the **Appendix**.



By above reasoning, we see that indeed $N(4,4;2) = 18$. ■

2 Problem 2

Problem 2.1. Prove that for all integers $r \geq 1$, there is a minimal number $N(r)$ with the following property. If $n \geq N(r)$ and the integers in $\{1, 2, \ldots, n\}$ are colored with r colors, then there are three elements x, y, z (not necessarily distinct) with the same color and $x + y = z$. Determine $N(2)$ and show that $N(3) > 13$.

Solution:

Proof. We take the vertex set as $\{1, 2, \ldots, n\}$, and proceed the coloring rule as follow: $\forall$ integer $i, j \in [1, n]$, if $i < j$ we color the set $\{i, j\}$ by the color of $(i - j)$. In particular we see in this construction, all vertex pair are colored in one of the r color as required. We claim that the minimal number $N(r)$ we want to find is actually:

$$N(r) =: N(\underbrace{3, 3, \ldots, 3}_{r\ 3s}; 2)$$

where the right hand side is the Ramsey number. We now look into what happens when $n \geq N(r)$. By **Ramsey's Theorem**, when $n \geq N(r)$, there exists either a 3-subset where all of its 2-subset are colored in the first color, or the second color, or the third color, ..., or the rth color. And let's assume that $x < y < z$, with $\{x, y, z\}$ be such existed subset. By assumption, we see $\{x, y\}, \{y, z\}, \{x, z\}$ are colored in the same color. We then further assume:

$$a = y - x$$
$$b = z - y$$
$$c = z - x$$

By our construction on the rule of coloring the set $\{i, j\}, i, j \in \{1, 2, \ldots, n\}$, we see that a, b, c are also colored in the same color. Then observe that:

$$\begin{aligned} & a+b=(y-x)+(z-y)=z-x=c \\ \iff & a+b=c \end{aligned}$$

thus we see that our construction holds with the requirement, proof done. ■

Determine $\mathbf{N(2)}$?

Suppose we color $\{1,2,\ldots,n\}$ by red and blue, so we proceed the coloring step by step. First we color 1 with red, then by $1+1=2$, we see we have to color 2 in blue, for 3, both works, so we color it in blue since if we color it in red, we left no choice for 4. For 4 since $2+2=4$, we cannot color it in blue, so we can only color it in red. But then we see that we left no choice of color for 5, since $1+4=5$ with 1 and 4 both colored in red and $2+3=5$ with 2 and 3 both colored in blue. Thus we see that $\mathbf{N(2)=5}$. A visualization can be seen as follows, with **R** denote as colored in red and **B** denote as colored in blue:

1	2	3	4	5
R	B	B	R	~~R/B~~

Proof that $\mathbf{N(3)>13}$.

Proof. We just need to find a counter example for $\{1,2,\ldots,13\}$, which with three color **R, G, B**, such that any $x<y<z$ with same color, we always have $x+y\neq z$. A counter example can be illustrate as follows, with **R, G, B** denote as colored in R, G, B respectively.

1	2	3	4	5	6	7	8	9	10	11	12	13
R	G	G	R	B	B	B	B	B	R	G	G	R

so our proof is done. ■

3 Problem 3

Problem 3.1. Let m be given. Show that if n is large enough, every $n\times n$ $(0,1)$-matrix has a principal submatrix of size m (i.e., a submatrix obtained by removing $n-m$ rows and the same $n-m$ columns), in which all the elements below the diagonal are the same, and all the elements above the diagonal are the same.

Solution:

Proof. We shall show that when $n>N(N(m,m;2),N(m,m;2);2)$, we will have our intended submatrix exists.

We can let M be such $n\times n$ $(0,1)$-matrix. And such M will defined a colored complete graph K_n with vertices labeled as $\{1,\ldots,n\}$ by the following rules:

- If $i<j$ for $i,j\in[1,n]\cap\mathbb{Z}$, color the edge red if $M_{ij}=1$.
- Otherwise, color the edge blue if $M_{ij}=0$.

By **Ramsey's Theorem**, since $n>N(N(m,m;2),N(m,m;2);2)$, there exists a monocromatic subgraph $K_{N(m,m;2)}$ with $N(m,m;2)$ vertices. Then we see the submatrix M' induced by this subgraph is either attains its upper triangular matrix attains all entries to be the same value or the lower triangular matrix attains all entries to be the same value.

We now put our attention to these $N(m,m;2)\times N(m,m;2)$ matrix. We shall defined a new coloring rules on $K_{N(m,m;2)}$ induced by this matrix as follows:

- If $i < j$ for $i, j \in [1, N(m, m; 2)] \cap \mathbb{Z}$, color the edge red if $M_{ji} = 1$.
- Otherwise, color the edge blue if $M_{ji} = 0$.

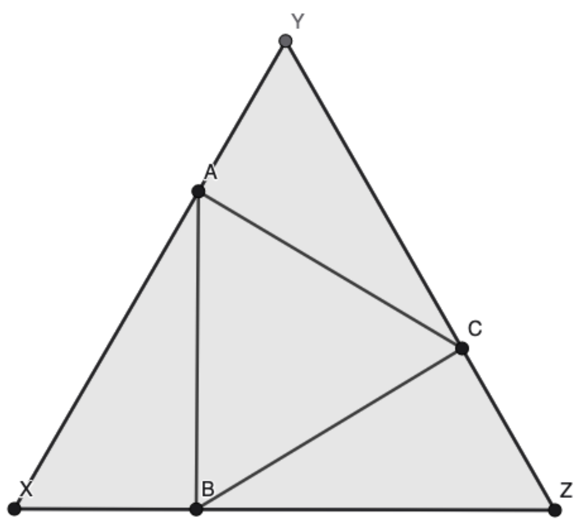
By **Ramsey's Theorem**, since $n > N(m, m; 2)$, there exists a monocromatic subgraph K_m with m vertices, and we shall see that submatrix M''induced by this subgraph attains all elements below the diagonal are the same and all the elements above the diagonal are the same by definition of our coloring rule, and such M'' is a principal submatrix of M' and M' is a principal submatrix of M, thus M'' will be a $m \times m$ principal submatrix of M which is intended. So we have done our proof. ■

4 Problem 4

Problem 4.1. Let XYZ be an equilateral triangle, and let S be the set of all points on the three segments XY, YZ, XZ. Prove that in any two-coloring of S, we can find a right-angled triangle all of whose vertices have the same color.

Solution:

Proof. We first extract out the inscribed equilateral triangle of the equilateral triangle XYZ, and then label the vertices of such triangle as ABC, as shown in the picture below:



Suppose that the points of the equilateral triangle is colored in red and blue. We shall analyze the case of the coloring of ABC:

- **ABC are colored in same color:** We see that a valuable property induced by the inscribed equilateral triangle ABC is that:

$$AB \perp XZ$$
$$AC \perp XY$$
$$BC \perp YZ$$

  Without losing generality, we suppose that point ABC are all colored in red. Then if any point on the line segment of the outer equilateral triangle other than ABC are colored in red, we can always find a right-angled red triangle. But if all of the point on the line segment of the outer equilateral triangle other than ABC are in blue, we can easily find other right-angled triangle blue triangle. $\longrightarrow$ **DONE!**

- **ABC are not all colored in the same color:** By **Pigeonhole Principle**, we see that two of the three points will be colored in the same color. Without losing generality, we suppose that AB will be colored in red, and C will be colored in blue. Then any point on the line segment XZ other than B is in red will resulted in the intended right-angled red triangle. But if all of the points on the line segment XZ other than B are colored in blue, then we can draw a perpendicular line from C to XZ, and mark the foot of the perpendicular as E, we see that both C and E are colored in blue in this situation. And with all the point on line segment XZ other than B are colored in blue, then we see $\triangle CEX$ will be an intended right-angled blue triangle. $\longrightarrow$ **DONE!**

So we see in all cases, we are managed to find a right-angled triangle with all of whose vertices have the same color. So the proof is done. ■

5 Problem 5

Problem 5.1. Find the chromatic polynomial of the n-cycle C_n for $n \geq 3$. Find the chromatic polynomial of the n-wheel W_n (this is the graph obtained from C_n by adding a new vertex and joining it to all vertices of C_n).

Solution: Before we proceed our solution, we first proof two lemmas stated as follows:

Lemma 5.2. Given a graph $G = (V, E)$, and given $e \in E$, if e is not a loop, then its chromatic polynomial satisfy that:

$$\chi_G(k) = \chi_{G'_e}(k) - \chi_{G''_e}(k)$$

where G'_e denotes the graph induced by G deleting the edge e, and G''_e denotes the graph induced by G contracting the edge e.

Proof. If e is an edge of G, not a loop, we can then break the proper colorings of G'_e into two classes:

- Those in which the endpoints of e have different colors. Such case is the proper colorings of G, with the chromatic polynomial given as $\chi_G(k)$.
- Those in which the endpoints of e have the same colors. Such case is the proper coloring of G''_e, with the chromatic polynomial given as $\chi_{G''_e}(k)$.

So it follows that:

$$\chi_{G'_e}(k) = \chi_G(k) + \chi_{G''_e}(k)$$

which is equivalent to:

$$\chi_G(k) = \chi_{G'_e}(k) - \chi_{G''_e}(k)$$

■

Lemma 5.3. For a tree T on n vertices, its chromatic polynomial is given by:

$$\chi_T(k) = k(k-1)^{n-1}$$

Proof. We shall proceed the proof by induction on the number of vertices:

1. **Base case:** When $n = 1$, we see that there are k kinds of different colorings, which aligns with our original statement.

2. **Inductive case:** We give our inductive hypothesis as follows: Suppose that when $n = p \geq 1$, we have that $\chi_T(k;p) = k(k-1)^{p-1}$, we want to see that when $n = p+1$, we have $\chi_T(k;p+1) = k(k-1)^p$.

   So when $n = p+1$, the tree $T_{p+1} = (V_{p+1}, E_{p+1})$ now have its number of vertices as $p+1$. We choose certain vertex $a \in V_{p+1}$ and delete it from the tree, such that after the deletion, the graph is still a tree. In particular we see that before the deletion, T_{p+1} attains $p+1$ vertices and p edges, and after the deletion, since it is still a tree, it attains p vertices and $p-1$ edges, it means that the deletion only delete a vertex that only connect to one vertex of T_p. By **Inductive Hypothesis**, we see that the chromatic polynomial of T_p is given as $\chi_T(k;p) = k(k-1)^{p-1}$. For vertex a, if we insert it back into T_p it only need to have different color with the vertex it connect to, in particular it has $k-1$ number of choices on coloring. So we see that:

   $$\chi_T(k;p+1) = (k-1)\chi_T(k;p) = (k-1)k(k-1)^{p-1} = k(k-1)^p$$

   So we see the inductive case also holds.

By above reasoning, we successfully proof the lemma. ■

We now **claim** that:

$$\chi_{C_n}(k) = (-1)^n(k-1) + (k-1)^n, \ \forall n \geq 3 \tag{1}$$

and

$$\chi_{W_n}(k) = (-1)^n(k^2-2k) + k(k-2)^n, \ \forall n \geq 4 \tag{2}$$

And we shall proceed our proof:

Proof. We first proof **Equation** 1. By **Lemma** 5.2, we see that:

$$\chi_{C_n}(k) = \chi_{C_n'e}(k) - \chi_{C_n''e}(k)$$

where we see that for the case of C_n, we have:

$$C_n'e = T_n$$
$$C_n''e = C_{n-1}$$

Then we see that:

$$\chi_{C_n}(k) = \chi_{C_n'e}(k) - \chi_{C_n''e}(k)$$
$$\iff \chi_{C_n}(k) = \chi_{T_n}(k) - \chi_{C_{n-1}}(k)$$

By **Lemma** 5.3, we have that:

$$\chi_T(k) = k(k-1)^{n-1}$$

So we see in fact we have:

$$\chi_{C_n}(k) = k(k-1)^{n-1} - \chi_{C_{n-1}}(k)$$
$$\iff k(k-1)^{n-1} = \chi_{C_n}(k) + \chi_{C_{n-1}}(k)$$

And it is easy to see that when $n = 3$, we have a triangle, and to properly color it with k colors, the chromatic polynomial will be $k(k-1)(k-2)$. In particular, we have:

$$\chi_{C_3}(k) = k(k-1)(k-2)$$

So by:

$$\chi_{C_n}(k) + \chi_{C_{n-1}}(k) = k(k-1)^{n-1} \tag{3}$$

We can then deduce that:

$$\begin{aligned}
\chi_{C_n}(k) &= -\chi_{C_{n-1}}(k) + k(k-1)^{n-1}\\
\chi_{C_{n-1}}(k) &= -\chi_{C_{n-2}}(k) + k(k-1)^{n-2}\\
\chi_{C_{n-2}}(k) &= -\chi_{C_{n-3}}(k) + k(k=1)^{n-3}\\
&\vdots\\
\chi_{C_4} &= -\chi_{C_3}(k) + k(k-1)^3
\end{aligned}$$

And hence sum up all of the above by coefficient to delete intermediate terms, we can deduce that:

$$\begin{aligned}
\implies \chi_{C_n}(k) &= k(k-1)^{n-1} - k(k-1)^{n-2} + k(k-1)^{n-3} - k(k-1)^{n-4} + \cdots +\\
&\quad (-1)^{n-4}k(k-1)^3 + (-1)^{n-5}k(k-1)(k-2)\\
&= (-1)\sum_{i=3}^{n-1} k(k-1)^i(-1)^{i-n} + (-1)^{n+1}k(k-1)(k-2)\\
&= (-1)^{n+1}[\sum_{i=3}^{n-1} k(k-1)^i(-1)^i + k(k-1)(k-2)]\\
&= (-1)^{n+1}k[\sum_{i=3}^{n-1}(1-k)^i + (k-1)(k-2)]
\end{aligned} \tag{4}$$

We then denote that:

$$A = \sum_{i=3}^{n-1}(1-k)^i$$

Then we can deduce that:

$$\begin{aligned}
A &= (1-k)^3 + (1-k)^4 + \cdots + (1-k)^{n-1}\\
\implies (1-k)A &= (1-k)^4 + \cdots + (1-k)^{n-1} + (1-k)^n\\
\implies -kA &= (1-k)^n - (1-k)^3\\
\implies A &= \frac{(1-k)^3 - (1-k)^n}{k}
\end{aligned}$$

We then plug A back to the **Equation** 4, and deduce that:

$$\begin{aligned}
\chi_{C_n}(k) &= (-1)^{n+1}[-(1-k)^n + (1-k)^3 + k(k-1)(k-2)]\\
&= (-1)^{n+1}[k(k-1)(k-2) - (k-1)(k-1)^2 - (1-k)^n]\\
&= (-1)^{n+1}[(k-1)(k^2-2k-(k^2-2k+1)) - (1-k)^n]\\
&= (-1)^{n+1}[1-k-(1-k)^n]\\
&= (-1)^n(k-1) + (k-1)^n
\end{aligned}$$

And thus we see that:

$$\chi_{C_n}(k) = (-1)^n(k-1) + (k-1)^n,\ \forall n \geq 3$$

So we successfully proof **Equation** 1!

Now considering $\chi_{W_n}(k)$, and we have following **key observations**:

- W_n actually consists of $n+1$ vertices, which attains C_n as its subgraph.
- The proper coloring of W_n can be proceed by the following procedure:
  1. First use a random color from the k-color set to color the center of the wheel. Namely first color the vertex that connects to all the rest of the vertices.

2. then the rest of the vertices form a wheel, and they can be given a proper coloring by $k-1$ different colors.

Thus **by the key observation**, we can deduce that:

$$\begin{aligned} & \chi_{W_n}(k) = k\chi_{C_n}(k-1) \\ \implies & \chi_{W_n}(k) = k((-1)^n(k-2) + (k-2)^n) \\ & \qquad = (-1)^n(k^2-2k) + k(k-2)^n \end{aligned}$$

And thus we see that:

$$\chi_{W_n}(k) = (-1)^n(k^2-2k) + k(k-2)^n, \ \forall n \geq 4$$

So we successfully proof **Equation** 2! ■

6 Problem 6

Problem 6.1. Prove that any graph G has at least $\binom{\chi(G)}{2}$ edges.

Solution:

Proof. Suppose there exists a graph $G = (V, E)$ attains its edges less than $\binom{\chi(G)}{2}$, we then properly color G with (exactly) $\chi(G)$ numbers of colors. We denote the color set as C and we see $|C| = \chi(G)$. We arbitrarily choose two different colors out of the color set C and denote two color as $i, j \in C$, and we have in total $\binom{\chi(G)}{2}$ ways of choosing. Since we suppose that $|E| < \binom{\chi(G)}{2}$, by **Pigeonhole Principle**, there is the situation that $\forall x, y \in V$, who are colored in, say $i, j, i \neq j$ respectively, are not connected by any edges! Then it means we can discard either i or j and result in a lower color set, but this contradicts to the definition of $\chi(G)$ $\lightning$. ■

7 Problem 7

Problem 7.1. Let G be a n-vertex, triangle-free, simple planar graph such that $n \geq 3$. Show that $\#E \leq 2n-4$.

Solution: Before we proceed the proof we shall proof the a lemma first:

Lemma 7.2. Given $G = (V, E)$ to be a planar graph, and we denote $\#F$ to be the total number of faces of G, $\#E$ to be the total number of edges of G, we denote F_i for some integer $i \in [3, n]$ to be the i-gon face in G. Then we have:

$$\sum_{i=3}^{n} iF_i = 2\#E \tag{5}$$

and

$$\sum_{i=3}^{n} F_i = \#F \tag{6}$$

Proof. In a planar graph, every edges correspond to two faces, which stays along different sides of the edge. Thus we have:

$$\sum_{\text{faces}} \#(\text{sides of the faces}) = 2\#E$$

and by defnition we see:

$$\sum_{i\geq 3} iF_i = 2\#E$$

and since we only attains n vertices, so there exists an upper bound as n-gon, so we have:

$$\sum_{i\geq 3} iF_i = 2\#E$$

which is exactly **Equation** 5. And **Equation** 6 is obvious followed by the total number of faces should be the sum of each i-gon faces. ■

So we then proceed **the proof of the original statement**.

Proof. We will divide the case on whether G is a connected graph or not.

- G is a **connected** graph: Since G is a connected planar graph, we see that it satisfy the **Euler's Equation** as follows:

$$\#F - \#E + \#V = 2 \tag{7}$$

  Since G is a triangle-free graph, in particular it contains no triangle faces, we see that by **Lemma** 7.2, we have:

$$\begin{aligned} &\sum_{i=3}^{n} iF_i = 2\#E \\ \Longrightarrow &\sum_{i=4}^{n} iF_i = 2\#E \end{aligned} \tag{8}$$

  and:

$$\begin{aligned} &\sum_{i=3}^{n} F_i = \#F \\ \Longrightarrow &\sum_{i=4}^{n} F_i = \#F \end{aligned} \tag{9}$$

  By **Equation** 8, we see that:

$$\begin{aligned} 2\#E = \sum_{i=4}^{n} iF_i &\geq 4\sum_{i=4}^{n} F_i \\ &= 4\#F \end{aligned}$$

  By **Equation** 7, we see that:

$$\begin{aligned} 2\#E &\geq 4\#F \\ &= 4(\#E - \#V + 2) \\ &= 4(\#E - n + 2) \\ \Longleftrightarrow \#E &\leq 2n - 4 \end{aligned}$$

- G is **not a connected** graph: If G is not a connected planar graph, then for each connected component, we add one edge to connect all the connected component back to form a connected planar graph G'. Suppose that we add in total $c > 0$ edges to create such graph G'. Then we can apply our previous result and conclude that:

$$\begin{aligned} &E(G') = E(G) + c \leq 2n - 4 \\ \Longrightarrow &E(G) \leq 2n - 4 - c \leq 2n - 4 \end{aligned}$$

where $E(G)$ and $E(G')$ denotes the total edges in the graph G and G' respectively. So we then see that in this situation:

$$\#E \leq 2n - 4$$

still holds.

So we successfully prove the original statement! ■

8 Appendix

8.1 Source Code

Python source code that simulates the non-existence of blue K_4 and red K_4 in **Problem 1**.

```python
import matplotlib.pyplot as plt
import numpy as np
import itertools

n = 17

def get_edge_color(v1, v2):
    # Calculate the direct difference between the two points
    diff = abs(v1 - v2)
    # On a circular structure, the shortest distance is the minimum of diff and n-diff
    min_diff = min(diff, n - diff)

    if min_diff in {1, 2, 4, 8}:
        return 'red'
    else:
        return 'blue'

    # If the difference is not in these sets, there is no edge
    return None

angles = np.linspace(0, 2 * np.pi, n, endpoint=False)
x_coords = np.cos(angles)
y_coords = np.sin(angles)

plt.figure(figsize=(12, 12))
ax = plt.gca()
ax.set_aspect('equal', adjustable='box')
plt.axis('off')


# Iterate through all pairs of vertices (i, j) with i < j to avoid duplicate drawing
for i in range(n):
    for j in range(i + 1, n):
        # Get the color of the edge
        color = get_edge_color(i, j)
        # If a color is defined, draw a line segment connecting the two vertices
        if color:
            plt.plot([x_coords[i], x_coords[j]], [y_coords[i], y_coords[j]], color=color, alpha=0.7, linewidth=1.2)

# Draw black circles at the calculated coordinates to represent the vertices
plt.plot(x_coords, y_coords, 'o', color='black', markersize=12)
# Add a numerical label to each vertex, slightly offset for better visibility
for i in range(n):
    plt.text(x_coords[i] * 1.1, y_coords[i] * 1.1, str(i), fontsize=12, ha='center', va='center')

plt.title('K_17 Graph Coloring', fontsize=16)
# Save the generated plot to a file
plt.savefig("k17_graph.png")
print("Graph has been generated and saved as k17_graph_en.png")


# Check for the existence of a monochromatic K_4 subgraph
```

```python
print("\nChecking for monochromatic K_4 subgraphs...")

found_monochromatic_k4 = False
monochromatic_subsets_found = []

# itertools.combinations(range(n), 4) generates all possible 4-vertex subsets
all_4_vertex_subsets = itertools.combinations(range(n), 4)

# iterations
for subset in all_4_vertex_subsets:
    # Get all 6 edges within this 4-vertex subgraph
    edges = list(itertools.combinations(subset, 2))

    # Get the color of the first edge to use as a reference
    first_edge_color = get_edge_color(edges[0][0], edges[0][1])

    is_monochromatic = True
    # Check if the remaining 5 edges have the same color as the reference
    for i in range(1, len(edges)):
        edge = edges[i]
        if get_edge_color(edge[0], edge[1]) != first_edge_color:
            # If an edge with a different color is found, the subgraph is not monochromatic
            is_monochromatic = False
            break

    # If the loop completes and is_monochromatic is still True, we found one
    if is_monochromatic:
        found_monochromatic_k4 = True
        monochromatic_subsets_found.append(subset)

print("-" * 30)
if found_monochromatic_k4:
    print("Verification Result: Monochromatic K_4 subgraphs were found.")
    print(f"A total of {len(monochromatic_subsets_found)} monochromatic K_4 subgraphs were found.")
    print("Here are some examples:")
    # Print only the first 10 found combinations as examples
    for s in monochromatic_subsets_found[:10]:
        color_name = get_edge_color(s[0], s[1])
        print(f"  - Combination: {s}, Color: {color_name}")
else:
    print("Verification Result: No monochromatic K_4 subgraphs were found among all 4-vertex subsets.")
print("-" * 30)

plt.show()
```