

ZPC-30

Bikramjyot, Farhan, Ishaan, Rachit

February 7, 2024

Instructions

Please read the following instructions carefully before proceeding further:

1. The test is of 2 hours. It will end *sharp* at **8:00 pm**.
2. Relevant reading material for all the questions has been provided in the document itself.
3. If you are stuck on a question that you cannot figure out, move on. Nothing good ever comes out of being hung up on something.
4. Please feel free to contact the invigilators in case of any queries.
5. We will provide hints to questions in the last hour, depending on the participation and responses.
6. All the best. GL HF :)

CHROMATIC POLYNOMIALS

Note: We will only be working with simple undirected graphs (graphs which do not contain duplicate edges or self-loops).

1 Graphs

A graph can be defined as a pair $G = (V, E)$.

The elements of the set V are called the *vertices* of the graph, and the elements of the set E are called the *edges* of the graph. Every element of E is a 2-element subset of V .

The following are examples of graphs, represented both diagrammatically and in notation,

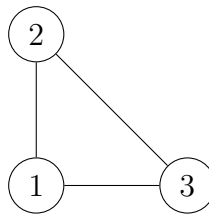


Figure 1: Graph $G = (E, V)$ where $E = \{1, 2, 3\}$ and $V = \{\{1, 2\}, \{2, 3\}, \{3, 1\}\}$

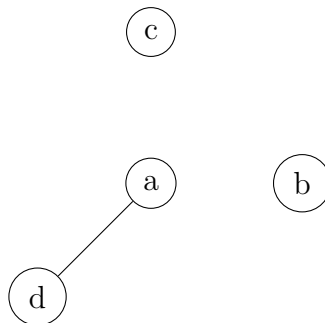


Figure 2: Graph $G = (V, E)$ where $V = \{a, b, c, d\}$ and $E = \{\{a, d\}\}$

Note that vertices of a graph may also be unlabeled:

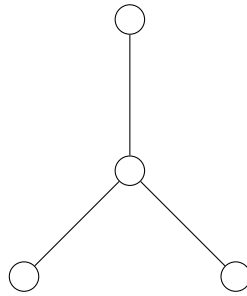


Figure 3: A graph without labels.

If we want to represent this graph formally, we can simply assign our own labels to nodes.

As we can see, graphs are useful in scenarios where we need to represent a set of objects with a sense of “connection” between some pairs of those objects.

We could, for example, make a graph whose vertices are the set of people at a party, and an edge between two vertices could represent pairs of people who already know each other.

Now we come to a popular area of research in graph theory, graph colorings.

2 Graph colorings

A *coloring* of a graph $G = (V, E)$ is a mapping $f : V \rightarrow C$ where C is a set of symbols or objects known as *colors*, where any two vertices sharing an edge must not be mapped to same *color*. More formally, for all $\{v_1, v_2\} \in E$, $f(v_1) \neq f(v_2)$.

Let’s consider an example.

Suppose we have a group of n students sitting in a circle. We want to assign each student a *team*, **red** (denoted by R) or **blue** (denoted by B), such that no two students sitting next to each other share the same team.

For any given n , we can represent this scenario as a graph; more specifically C_n , the **circular graph** with n nodes.

Consider $n = 8$.

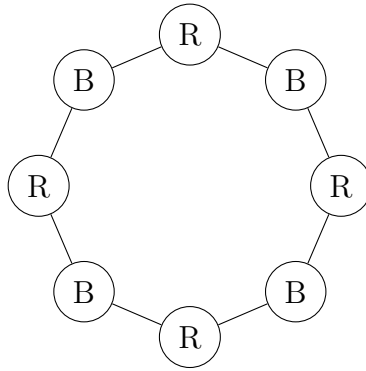


Figure 4: A 2-coloring of the graph C_8 .

The $n = 2k$ (even cycles) case is quite easy. We simply color one node red or blue, and keep filling all adjacent nodes with the opposite color.

However, when $n = 2k + 1$ (odd cycles), it is actually impossible to make a valid coloring using 2 colors.

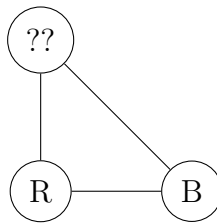


Figure 5: There exists no 2-coloring of graph C_3 .

As we can see in the figure above, an odd cycle of length 3 cannot be colored using 2 colors.

Convince yourself that no odd cycle can have a coloring with 2 colors.

However, all odd cycles can be colored using 3 colors. Let's introduce another color, G . An odd cycle of length 5 can be colored using R , B and G as follows,

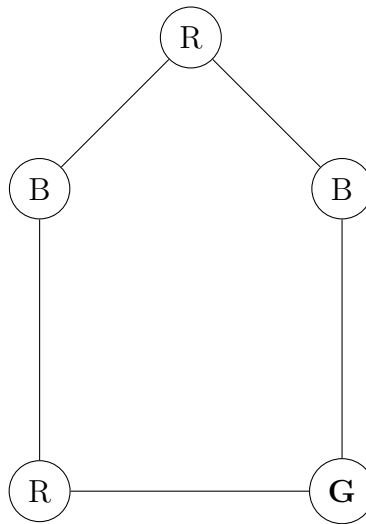


Figure 6: A 3-coloring of graph C_5 .

3 Chromatic Number

The minimum number of colors required to make a valid coloring of a graph G is called the **chromatic number** of G , often denoted as $\chi(G)$.

As we have seen, graphs of type C_{2k+1} (odd cycles) have chromatic number 3 ($\chi(C_{2k+1}) = 3$ for all k) and graphs of type C_{2k} (even cycles) have chromatic number 2 ($\chi(C_{2k}) = 2$ for all k).

For example, consider the complete graph on n vertices, K_n , in which all pairs of nodes have an edge between each other.

K_5 is illustrated below:

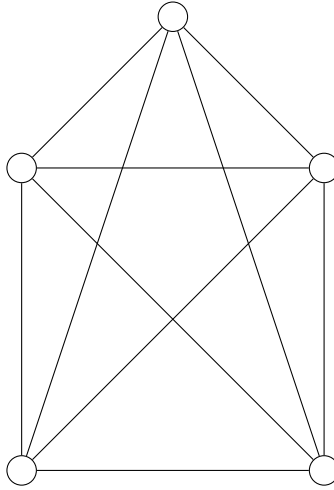


Figure 7: The complete graph of 5 vertices, K_5 .

Graphs of type K_n have chromatic number n , because all the nodes need to be of different colors.

Note that if a graph can be colored using k colors, it can also be colored using $k + 1$ or $k + 2$ (and so on) colors, because using all colors in the set of colors is not mandatory. Each node must have at least one color, so we cannot go lower than the chromatic number. Therefore, every valid k -coloring of a graph is also a valid l -coloring where $l > k$.

4 Chromatic Polynomials

What if we were given a graph G , and we wanted to construct a function which gives us the number of k -colorings of G for any given k ?

Turns out we have a solution to this, called the **chromatic polynomial** of the graph G .

The chromatic polynomial $p_G(\lambda)$ is a polynomial function in λ :

$$p_G(\lambda) = \lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \cdots + a_n$$

where $p_G(k)$ is the number of valid k -colorings of the graph G .

Consider the example of E_n , the graph with n nodes and *no edges*. What is the chromatic polynomial of this graph?

Suppose we have k colors. Each node has k choices of colors independent of each other, since there are no edges. So the number of k -colorings of E_n is k^n .

So, the chromatic polynomial for E_n is $p_{E_n}(\lambda) = \lambda^n$.

Now, consider the opposite of this graph, the complete graph K_n which we discussed earlier. **The chromatic polynomial of the complete graph K_n is:**

$$p_{K_n}(\lambda) = \lambda(\lambda - 1)(\lambda - 2) \cdots (\lambda - n + 1)$$

Here's how we came up with the chromatic polynomial of K_n :

- Since for $k = 1, 2, \dots, n - 1$ there are no valid colorings, those k must be zeroes of the polynomial.
- Suppose $k \geq n$. Suppose we color the first node with color c_1 with λ choices. Then, the next node must be colored c_2 with $\lambda - 1$ choices, and so on, which gives us the expression above that *happens to have* numbers from 1 to $n - 1$ as its zeroes.

We'll also look at a more formal and objective way to calculate the chromatic polynomial of a graph in the upcoming section.

Make sure you understand everything by this point, ok? I know it's a lot to take in if you have never dabbled in graph theory, but feel encouraged to ask a coordinator anything you're unsure about. After all, the point is to learn something and have fun doing it!

5 Deletion-Contraction Algorithm

The deletion-contraction algorithm is an algorithm to compute the chromatic polynomial of a given graph.

Before we cover the formula, we need to introduce 2 new graph operations, **edge deletion** and **edge contraction**.

Edge deletion

Suppose $G = (V, E)$ is a graph with a non-empty edge set. Let $e \in E$ be an edge of the graph G . The deletion $(G - e)$ is defined as the graph $G = (V, E - e)$.

Essentially, we are removing the edge e from the graph without affecting any other edges.

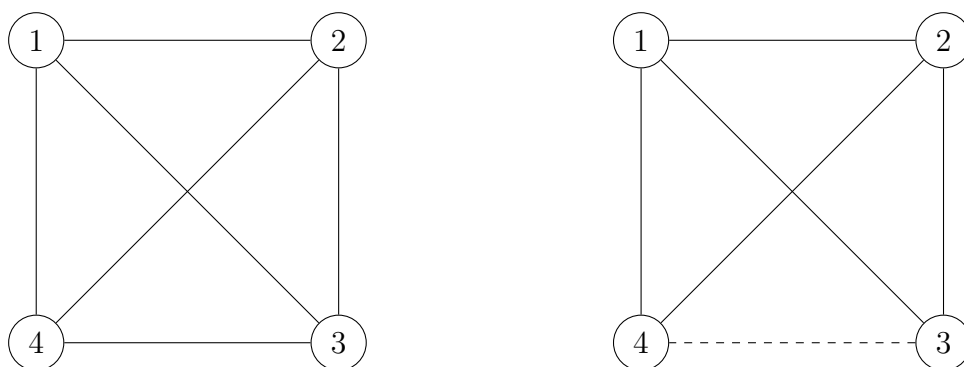


Figure 8: On the left: Graph K_4 . On the right: $(K_4 - \{3,4\})$. (To be painfully precise: $\{3,4\}$ is not contained in the edge set of the right graph.)

Edge contraction

Suppose $G = (V, E)$ is a graph with a non-empty edge set. Let $e \in E$ be an edge of the graph G . The contraction $(G \setminus e)$ can be seen as “merging” the two vertices v_1 and v_2 connected by the edge into one vertex as follows:

1. The edge v_1, v_2 is removed from the edge set.
2. Any edge e' where $v_2 \in e'$ is converted to $((e' - \{v_2\}) \cup v_1)$.
3. The node v_2 is removed from the graph.

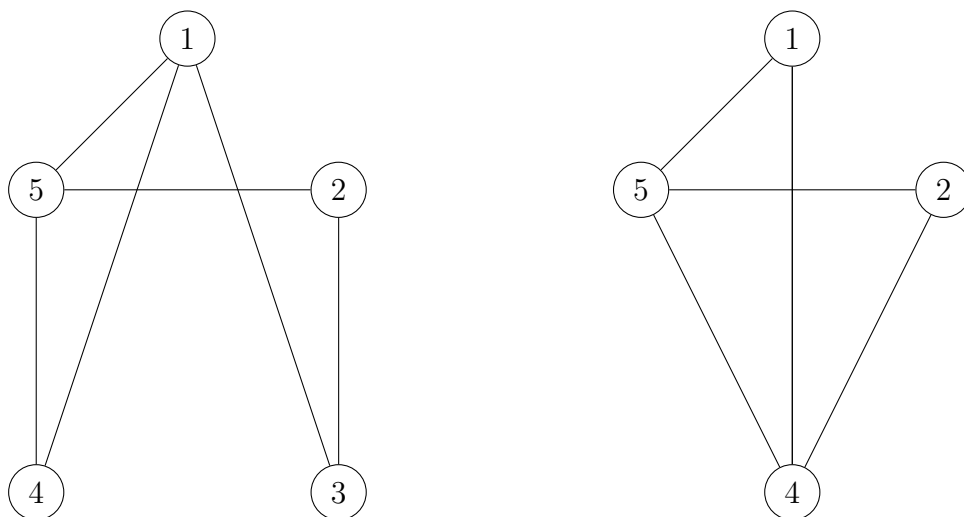


Figure 9: On the left, an arbitrary graph G . On the right, $(G \setminus \{3, 4\})$. Nodes 3 and 4 are merged into 4.

Statement of the Deletion-Contraction Algorithm

Suppose we need to calculate the chromatic polynomial of a graph $G = (V, E)$.

If $E(G) = \emptyset$, then G is the empty graph of $|V|$ vertices, and its chromatic polynomial is $p_G(\lambda) = \lambda^{|V|}$.

Otherwise, we inductively calculate the chromatic polynomial of G as follows:

$$p_G(\lambda) = p_{G-e}(\lambda) - p_{G \setminus e}(\lambda)$$

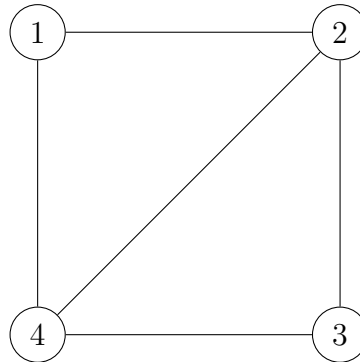
Where e is any edge in G .

Intuition behind the formula

- Any valid coloring of G is a valid coloring of $(G - e)$ (convince yourself why.)
- The only colorings of $(G - e)$ which are not valid in G are the colorings of $(G \setminus e)$, when the color of v_1 and v_2 is the same in $G - e$. (convince yourself why.)

6 Problems

1. Given graph G below:



- (a) Compute the chromatic polynomial of G . 3 points
- (b) Compute $\chi(G)$. 1 point
- (c) Compute the number of 5-colorings of G . 1 point
2. Prove the following by induction on n :
- (a) $p_{C_n}(\lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$, where C_n is the cycle of n vertices.
(Ask the coords or look at the theory given for an example of C_n if you're confused) 4 points
- (b) If T_n is any *tree* of n nodes, $p_{T_n}(\lambda) = \lambda(\lambda - 1)^{n-1}$ (a tree of n vertices is a graph with $n - 1$ edges with no cycles, and every node is an endpoint of at least 1 edge.) 4 points
3. Prove that for any $n \geq 4$, two non-isomorphic graphs of n nodes (graphs which cannot be the same if you rename the nodes) may have the same chromatic polynomial. (An explicit construction of both graphs is enough, you do not need to prove that they are not isomorphic.) 3 points
4. Suppose you already know the chromatic polynomial of a graph with n nodes, and its coefficients are stored in an array of size $n + 1$. Write an algorithm to compute the chromatic number of the graph in $O(n \log n)$ time (you don't

need to be too formal but there should be no ambiguity.)

4 points

5. The *Mycielskian* $\text{My}(G)$ of a graph $G = (V, E)$ is defined as follows:

- Let $V = \{v_1, v_2 \cdots v_n\}$ and $E = \{e_1, e_2, \cdots e_m\}$.
- Let $W_1 = \{w_1, w_2 \cdots w_n\}$ (a new vertex set of size n)
- Let $E_1 = \bigcup_{e=\{v_i, v_j\} \in E} (\{\{v_i, w_j\}, \{w_i, v_j\}\})$
- Let $W_2 = \{x\}$ (another new vertex set of size 1)
- Let $E_2 = \{\{w_i, x\} | w_i \in W_1\}$
- $\text{My}(G) = (V \cup W_1 \cup W_2, E \cup E_1 \cup E_2)$

What can you say about $\chi(\text{My}(G))$ if you know $\chi(G)$? State the relation between both and reason why (again, no need to be too formal).

15 marks

□□□□□□□□□□□□□□