Computational Graph Theory

Hari Desikan, typeset by Pranav Konda

December 21, 2020

Contents

1 Some Relevant Definitions

Let's define some terms before we proceed.

Definition 1 (Vertices and Edges)**.** *A* **node**, *or* **vertex** *is what is used to construct a graph. They are connected by* **edges**. *Two vertices are* **adjacent** *if they are connected by a single edge. The* **degree** *of a vertex is the number of edges connecting that vertex to another.*

Definition 2 (Graphs)**.** **Graphs** *are sets of nodes and the edges which connect them. A* **labelled** *graph is one in which each node is labelled, and an* **unlabelled graph** *is one in which no nodes are labelled. A graph is* **directed** *if the edges are oriented, and* **undirected** *if the edges are unoriented. Two graphs are* **isomorphic** *if the vertices of one can be relabelled to form the other.*

Definition 3 (Types of Graphs)**.** *A graph is* **planar** *if it can be drawn such that no two edges intersect. A graph is* **simple** *if no two vertices are connected by more than one edge and if no vertex is connected to itself. An unlabelled graph is* **connected** *if you can get from any vertex to any other vertex by travelling along edges. A* **cycle** *is a walk from a vertex back to itself which does not use any edge more than once. A* **tree** *is a connected graph with no cycles.*

2 Euler's Formula

This is really the only computational theorem that you need to know.

Theorem 1 (Euler's Formula)**.** *In a connected planar graph G with F faces, V vertices, and E edges, we have that*

$$F + V = E + 2.$$

Let's look at a quick example:

Example 1 (Volume 2, AoPS)**.** *If a connected planar graph with v vertices, each of degree 4, has 10 faces, find v.*

Solution 1. *Use Euler's Formula. We have*

$$f + v = e + 2 \implies 10 + v = e + 2.$$

Now note that the sum of the degrees of the vertices is twice the number of edges to arrive at

$$10 + v = 20 + 2 \implies \boxed{v = 12},$$

and we are done.

Remark 1. *Note that a similar result does hold for polyhedra in general, and this is known as the* **Euler characteristic**. *For a polyhedron with V vertices, E edges, and F faces, we have*

$$V - E + F = \chi,$$

where χ is Euler characteristic. For a convex polyhedron in 3 dimensions, $\chi = 2$. For nonconvex polyhedra, multiple possible Euler characteristics exist. While originally defined for polyhedra, the Euler Characteristic has many uses in homological algebra and other advanced fields. Proving this assuming the planar graph result holds is useful (and the interested reader is encouraged to attempt this) but the general proof is rather axiomatic and is thus omitted.

3 Cycles

The appearance of a function from some finite set of elements to themselves is often a huge indicator that the problem should be, at the very least, modelled by a graph.
For example, if a problem reads "A function $\{1, 2, \ldots, 100\} \to \{1, 2, \ldots, 100\}$ has the property that...," it is undoubtedly a cue that the numbers $\{1, 2, \ldots, 100\}$ should be modelled by vertices and that there should be a directed edge from x to $f(x)$ for all x in the domain of f.

Example 2 (HMMT November 2015/10)**.** *Let N be the number of functions f from $\{1, 2, \ldots, 101\} \to \{1, 2, \ldots, 101\}$ such that $f^{101}(1) = 2$. Find the remainder when N is divided by 103.*

Solution 2. *Modelling the integers as vertices leaves us with a walk of length 101 starting on one vertex and ending on another. Motivated entirely by the fact that many graph theory problems deal with cycles rather than walks, we instead consider functions satisfying $f^{101}(1) = 1$. Finding the number of such functions with such a restriction is exceedingly common.*
If 1 is in any cycles, the cycle length must divide 101 (or else in 101 steps, you will not end back up at 1 when starting a walk from vertex 1.) But only 1 and 101 divide 101. So the number of functions with $f^{101}(1) = 1$ is the number of distinct labelled graphs that are isomorphic to C_101 (the cycle graph on 101 vertices) plus the number of functions where 1 maps to itself. This leaves us with $100! + 101^{100}$ functions.
We now make a clever observation: the number of functions such that $f^{101}(1) = 2$ is the same as the number of functions such that $f^{101}(1) = 3$, or $f^{101}(1) = 101$. The output 2 is arbitrary.

Consider

$$\sum_{i=1}^{101} N(i)$$

with N(i) being equal to the number of functions such that $f^{101}(1) = i$. Clearly, the sum is 101^{101} as this is just all functions from 1...101 to itself with no restrictions.

$$\sum_{i=2}^{101} N(i) = 100N(2) = \sum_{i=1}^{101} (N(i)) - N(1)$$

, where the first equality is a consequence of the first block of text on this past and the equality of the first and third terms a simple sum manipulation.

But we know the last term! It is $101^{101} - (100! + 101^{100})$!

This means $100N(2)$ also takes on this same value, or

$$100N(2) = 101^{101} - (100! + 101^{100})$$

$$N(2) = \frac{101^{101} - 100! - 101^{100}}{100} = 101^{100} - 99!$$

It remains to find $N(2) \mod 103$, but this is not a graph theoretical step so much as it is a number theoretical one. We outline the steps in case the watcher has not watched the relevant CNCM lectures: $101^{100} - 99! \equiv -2^{-2}$ *(Euler's Totient, negation)* $-\frac{101!}{(-2)(-3)} \equiv \frac{1}{12}$ *(Wilson's)*$\equiv \boxed{43}$ *(Euclidean Algorithm)*

4 A Brief Look at Tournaments

Sometimes, it is valuable to model tournaments (and related scenarios) with large numbers of teams as graphs. Most of the time, if you are given characteristics of a tournament that are well modelled by a graph (such as a very deterministic win-loss record for various teams) you will be able to arrive at a useful model. This is because in tournaments, two teams compete against one another in various stages; this sort of competition is very analogous to undirected and directed edges, depending on information given. However, it is important to remember that there are many tournaments that are not-so-easily modelled by graphs. Problem writers can easily draw many diverse scenarios that are tangentially related to tournaments.

5 Advanced Exercises with Trees

These exercises might require advanced knowledge and/or non-computational techniques to solve, and are best left for readers already well-versed in graph theory. The alternate definitions of trees are somewhat important, but these concepts are more likely to show up on the AMC tests where IDing and using common properties is useful.

Definition 4 (Tree 2)**.** *A tree is* also *a connected graph on n nodes with $n-1$ edges.*

Remark 2. *Try to prove this to yourself somewhat rigorously- it will familiarize yourself with the type of casework present in many graph theory problems.*

Definition 5 (Tree 3)**.** *A tree is* also *a connected graph with 1 face.*

Problem 1 (Classic)**.** *Find the number of full nonisomorphic binary trees with n nodes. Each node of a full binary tree has 1 incoming edge and 0 or 2 outgoing edges save the uppermost node, which has 0 incoming edges.*

Remark 3. *The solution to this problem is well-known, and can be found on various sites. The problem itself would never show up on a contest, but ideas derived from working through the problem do show up in a few advanced computational contests (like the OMO.)*

Problem 2 (HMMT November 2020 Guts Round)**.** *In a single-elimination tournament consisting of $2^9 = 512$ teams, there is a strict ordering on the skill levels of the teams, but Joy does not know that ordering. The teams are randomly put into a bracket and they play out the tournament, where the better team always beats the worse team. Joy is then given the results of all 511 matches and must write a list of teams such that she can guarantee that the third best team is on the list. What is the minimum possible length of Joy's list?*

6 Related Topics and Resources

Graph Theory is a very large field. Here are a few topics that you might look into if you are interested in graph theory at a level that is applicable to various contests:

1. Eulerian Graphs (and Euler's corresponding theorem with constructive proof)
2. Adjacency Matrices (and powers, generating functions thereof)
3. Chromatic Numbers and Polynomials
4. Depth First Search (and other ways to methodically, efficiently search graphs)
5. Dijkstra's Algorithm (and Prim's Algorithm)

If you seek to train rigorously in Olympiad Graph Theory, the relevant Graph Theory chapter in Pranav A Sriram's combinatorics book is a powerful, free resource to learn many theorems that only begin to show up at the olympiad level.