Discrete Mathematics

Chris Harrison

November 18, 2020

Abstract

This is an abstract.

Introduction

In this course we will start with some preliminaries (lesson 1), including sets (s. 1.1) and the pigeonholeprinciple (s. 1.2), before moving on to an introduction to graphs (lesson 2), trees (lesson 3) and planar graphs (lesson 4).

1 Preliminaries

1.1 Sets

1.2 Pigeonhole Principle

2 Graphs: An introduction

3 Trees

4 Planar Graphs

4.1 Bipartite Graphs

4.2 Hamiltonian Cycles

4.3 Planar Graphs

Euler's Relationship for any connected planar graph

$$n - m + r = 2$$

(Th10.2, p190)

If G is planar then order, i.e. number of vertices, $n \geq 3$, size m, then $m \leq 3n - 6$

The boundary of every region must have at least 3 edges.

This is obvious for $n = 3$, so $n \geq 4$ and $m \geq 3$. Suppose G is planar and has r regions.

$$n - m + r = 2$$

Let's sum up all the arcs touching all the regions, N

$$3r \leq N$$

Each arc can only touch two regions

$$N \leq 2r \tag{1}$$
$$2 = n - m + r \tag{2}$$
$$6 = 3n - 3m + 3r \tag{3}$$

but

$$3r \leq 2m$$

So

$$6 \leq 3n - 3m + 2m \tag{4}$$
$$6 <= 3n - m \tag{5}$$
$$m <= 3n - 6 \tag{6}$$
$$\tag{7}$$

We can go on to say if G is a graph of order at least 3 and size m, $m > 3n - 6$, G is nonplanar. For K_5 that works.

$K_{3,3}$ it doesn't.

Proof that $K_{3,3}$ is not planar:

Using Euler $A = 9$, $N = 6$, so R must be 5.

Pick a region, its boundary must be a cycle and (it is bipartite) it must be an even cycle, at least 4, perhaps 6.

5 regions, $5 * 4$ arcs, each arc separates 2 regions so $(5 * 4/2) = 10$ arcs required ... but we only have 9.

Contradiction

4.4 Ore's Theorem

Let G be a (finite and simple) graph with $n \geq 3$ vertices. We denote by deg_v the degree of a vertex v in G, i.e. the number of incident edges in G to v. Then, Ore's theorem states that if

$deg_v + deg_w \geq n$ for every pair of distinct non-adjacent vertices v and w of G

then G is Hamiltonian.

4.4.1 Proof

It is equivalent to show that every non-Hamiltonian graph G does not obey the condition.

Accordingly, let G be a graph on $n \geq 3$ vertices that is not Hamiltonian, and let H be formed from G by adding edges one at a time that do not create a Hamiltonian cycle, until no more edges can be added.

Let x and y be any two non-adjacent vertices in H. Then adding edge xy to H would create at least one new Hamiltonian cycle, and the edges other than xy in such a cycle must form a Hamiltonian path $v_1 v_2 ... v_n$ in H with $x = v_1$ and $y = v_n$.

For each index i in the range $2 \leq i \leq n$, consider the two possible edges in H from v_1 to v_i and from v_{i-1} to v_n. At most one of these two edges can be present in H, for otherwise the cycle $v_1 v_2 ... v_{i-1} v_n v_{n-1} ... v_i$ would be a Hamiltonian cycle.

Thus, the total number of edges incident to either v_1 or v_n is at most equal to the number of choices of i, which is $n - 1$. Therefore, H does not obey the property, which requires that this total number of edges ($deg_{v_1} + deg_{v_n}$) be greater than or equal to n.

Since the vertex degrees in G are at most equal to the degrees in H, it follows that G also does not obey the property.

[1]

[1] Based on the proof outlined in https://en.wikipedia.org/wiki/Ore%27s_theorem.

5 Algorithms

Algorithsm are a process or rules to follow. They should:

- take an input, give an output. For example ...
- be deterministic, i.e. for the same input the output will always be the same, cf functional programming
- and finite, cf computability.

They could be recursive, then they need a base case and recursive step. Typically the recursive step makes the problem smaller, converging on the base case.

An algorithm could:

- involve **heuristics**, finding decent solutions quickly, but not sure to be optimal
- be greedy, so exploit strong local solutions, with no regard to the bigger picture, e.g. hill-climbing
- ad-hoc, an impromptu strategy, unique to a particular problem, hard to generalise.

5.1 Flowcharts

Shapes

5.2 Orders of Growth

Order of growth - worst case, proportional to some function of the input size

Comparison is done in extreme cases, e.g. n log n vs n2

Offline vs online problems

Bubble sort, then shuttle Quick sort

Packing : next fit, first fit, first fit decreasing, full bin

Bin Packing - as a decision problem - is NP Complete

Knapsack problem - portfolio management,