Graph Theory

First Definitions and Eulerian Graphs

This session aims to give an introduction to the field of graph theory. This field concerns itself with objects called graphs, which are very useful to modelize all sorts of common behaviors (maps, social interactions, logical reasoning, neural networks).

A graph is an object composed of two parts : a set of vertices – also called points or nodes– and a set of edges – lines. Each edge is actually given as a pair of vertices. This terminology is justified by the fact that any polygon is indeed a graph (and in a way, polyhedras also are). We generally denote the number of vertices by n and the number of edges by m. As defined here a graph is a very simple object, but it still has a wealth of interesting problems. Consider the following very simple problem – and first problem ever studied in graph theory – called the bridges of Königsberg.

The figure above shows a city, and the question is whether we can cross each bridge exactly once. It appears impossible, and we can prove it is the case. To do so, we can modelize the city as a graph, where each zone is a vertex, and bridges are edges between vertices. The question then becomes “is it possible to find a path that goes exactly once through each edge ?”. This is in turn equivalent to “is it possible to draw the graph without raising your pen?”, just like in the puzzle where one has to draw a house. Graphs that have this property are called Eulerian, and it turns out that there is a very simple set of conditions equivalent to being Eulerian :

- The graph has to be connected (there is a path from any vertex to any other).
- Every vertex has to have even degree – the degree being the number of edges around that vertex – except at most two vertices.

The first condition is trivial, but why is the second necessary ? Well, what happens when you have a vertex with odd degree ? Each time you get to it you need to leave, which uses two edges. The only way to use all edges one time each is then to start from that vertex, or finish on it. Hence there are at most two vertices with odd degrees. To find the path going through all edges there is a simple method called Fleury’s algorithm and a faster one called Hierholzer’s algorithm.

Remark. We can observe that there are either zero or two vertices with odd degrees. To prove this, consider the number of edges m. Each edge adds one to the degree of two vertices. Hence the sum of degrees must be even, so there can’t be an odd number of vertices with odd degrees. This method is called double counting.

Remark. There is a related notion, called a Hamiltonian graph, where one can find a path going exactly once through each vertex (but can ignore many edges). Although finding an Eulerian path can be done in linear time (hence in $O(m)$), finding a Hamiltonian path is very hard (the problem is $\mathbf{NP-complete}$, and has no algorithm working in sub-exponential time).

Common Types of Graphs

The simplest graph is the line, which starts with a single vertex and has an edge to the next and so on until we reach the last vertex, never going twice through a vertex. We can enrich this slightly by adding an edge between the first and second vertices, creating a cycle graph – by extension, any path going back to itself is called a cycle.

A second common type of graph is the tree. A tree has a root, internal vertices, and leaves, and vertices in it have parents and children (a is an child of b if the path from b to the root goes through the edge ab). Trees can be defined by multiple equivalent properties :

- A connected graph with $n - 1$ edges
- A connected graph with no cycles
- A graph where there is exactly one path going from one point to another.

Finally, a graph on n vertices where every one of them has an edge to every other (hence a total of $\frac{n(n-1)}{2}$ edges) is called a complete graph or a clique.

Usual Properties

Graphs have many properties, and we can look at distances between pair of points (in the number of vertices on the shortest path from one to another). If we are on a line (which is a special kind of tree), the extremities are at distance $n-1$. On a cycle, any two points are at distance $\leq n/2$. However, if one vertex has an edge between it and any other vertex, the distance would be two. This type of graph is called a star. Stars are very nice because the distances are small, but they have a huge number of edges around one point which is awkward for real life purposes (if you consider the edges to be connections between computers, you really don't want to have one computer connected independently to all others on the internet. If we limit the number of edges on any vertex to k, the distance between pairs of points increases (it is normal, because we have at most k neighbours (vertices at distance 1), k^2 vertices at distance 2, and k^i at distance i. By simply adding all possible vertices at distances 1,2,3...i, we get a tree where all the nodes (except the last level) have degree k.

That kind of tree, called a complete binary tree when $k=3$ (except the root who has 2 children), has 2^i children at distance i. Hence, the distances are still small between two points (at most $2\log_2 n$) while keeping small degrees everywhere. Real networks often have tree-like structures, with additional redundancy in case a vertex is removed.

As mentioned, being connected means that one can go from any point to any other. A connected component is a set of vertices where we can go from any to any other (hence being connected means having one connected component). Trees and cycles are connected graphs. We can also look at a generalization of connectivity, called k-connectivity, where a graph is k-connected if and only if we need to remove k vertices to disconnect it. Trees are 1-connected, whereas cycles are 2-connected.

We can also introduce many other specialisations of graphs : adding numbers on edges can be used to have maps (the numbers are road lengths), or potential water networks (the numbers are the cost needed to build a pipeline, or a pipeline's capacity). We can also orient the edges, which can then only be traversed in one way. Or we can add probabilities on each vertex to simulate someone choosing a path randomly (those are commonly called Markov chains).

We can look at one last problem before going to the main problem studied in this session : the friends and strangers problem. We have a group of n people, and we wonder if we have one of the following :

- A group of k_1 people who all know each other.
- A group of k_2 people where no-one knows anyone else.

If $k_1 = 1$ or $k_2 = 1$, it is always true. If $k_1 = k_2 = 2$, it is true as long as we have two people. The problem arises when $k_1 = k_2 = 3$, and can be modelized easily with graph theory : we consider a set of n vertices, and we add an edge between two vertices if they know each other. Then the question becomes : do we have a triangle or an anti-triangle ? . It is easy to create graphs with 5 vertices not satisfying that property, but as soon as we have 6 we always have one of them. Let's take any vertex and prove it : either it has degree ≥ 3, or degree ≤ 2. If it has degree ≥ 3 and any two vertices among its neighbours share an edge, we have a triangle. If they don't, we have an anti-triangle. By symmetry it is also true if it has degree ≤ 2 (replacing triangle by anti-triangle). It turns out that this method works quite well to bound the number of people we need so that any graph on that number of people (written $R(k_1, k_2)$) satisfies one of the two conditions. By taking a vertex and looking at its neighbours, we can show that $R(k_1, k_2) \leq R(k_1 - 1, k_2) + R(k_1, k_2 - 1)$. This bound is easy, but finding the exact value seems an extremely hard task (we don't know the value of $R(5,5)$ yet, and testing for all graphs smaller than the bound is not possible as the order of magintude for the number of graphs to test is $2^{2^{2(k_1+k_2)}}$).

Minimal Separators

In a graph, a separator is a subset of nodes that disconnects two points that were previously connected. This has many applications in practice because many divide-and-conquer algorithms that seek to cut graphs in smaller parts depend on the number of separators. The number of separators is smaller than the number of subsets of nodes (which is equal to 2^n), but it can still be huge (a line has $2^{n-2} - n + 1$ separators). This seems not really relevant, so we need to introduce a more interesting concept : a separator that contains no other separators (hence the term minimal).

This raises a question : you might have a separator such that no strict subset of it separates two nodes a and b, but such that a subset separates a from another node c. To handle this we will look at minimal (a,b)-separators (we'll call them separators for short). If a and b are the extremities of a line, we get $n-2$ (a,b)-separators, and if they are on two ends of a cycle we get $\theta(n^2)$, because each separator has one node on the top path, and one on the bottom path (hence about $n/2$ possibilities on each, and we take the product). There is a difference between minimal (a,b)-separators and minimal separators (which are subsets of points that are minimal (a,b)-separators for two non-specified points a and b). However, it is easy to see that the number of minimal separators is at most n^2 times the number of minimal (a,b)-separators. As we'll see that this number is exponential and that n^2 is small compared to that, we shall from now on talk about separators to mean minimal (a,b)-separators (as there isn't much difference in practice).

Our goal is now to find $S(n)$, the function giving the maximal number of separators in any graph of size n. We will show two bounds : a simple lower bound, and a slightly more complex upper bound. There exists a recent result showing an improvement on the lower bound in English which you can read at https://arxiv.org/abs/1503.01203.

First, let's consider two points a and b, and k paths of 3 nodes between them. Any minimal (a,b)-separator has exactly one node on each path (because if it didn't it would have a path going from a to b, and if it had two it wouldn't be minimal). Hence there are 3^k minimal separators. Hence there are $3^{\lfloor \frac{n-2}{3} \rfloor}$ minimal separators, or about 1.442^n in total (the best bound known is around 1.452^n). Hence $1.442^n \leq S(n)$.

Now, we'll try to show an upper bound on the number of minimal separators. To do so we'll find a function from the set of separators of size k in a graph of size n to the number of separators of size k or $k-1$ in a graph of smaller size. Let's then consider a separator made up of k vertices on a graph on n vertices. If we remove the separator we get at least two connected components around a and b, so let's take the smallest one (and by symmetry let's say it's around a, and has size n_a). Then we take a neighbour c of a (which might be in the separator). Either c is in the separator or it isn't. If it is, then by looking at the graph where it is removed we get a separator of size $k-1$ on a graph where n_a hasn't changed. If c isn't in the separator, then we can contract the edge ac, which means that we add an edge between a and all the neighbours of a that weren't neighbours of c. In this new graph, the size of the separator hasn't changed, but the size of the connected component has become $n_a - 1$ (this is because contracting an edge that isn't in the separator does not affect the fact that a and b are not connected).

This means that the number of separators of size k and with a smallest connected component of size n_a, which we will write $S(n, k, n_a)$ satisfies the following equation :

$$S(n, k, n_a) \leq S(n, k-1, n_a) + S(n, k, n_a - 1)$$

Let's now define $S'(k + 2n_a) = \max_{n_a \leq n/2, k \leq n} S(n, k, n_a)$. Then $S'(n) \leq S(n-1) + S(n-2)$, because for the maximal value,

$$S(n, k, n_a) \leq S(n, k-1, n_a) + S(n, k, n_a - 1) \leq S'(n-1) + S'(n-2)$$

Hence $S'(n)$ satisfies the same relation as the Fibonnacci function, and is at most $O(\phi^n - \bar{\phi}^n)$. However, $S(n) \leq S'(n)$, because $k + 2n_a$ is smaller than n. This means that the number of minimal separators is at most $\phi^n \approx 1.61^n$.

This bound is currently the best known upper bound, and conjectured to be correct (which is way more believable now that an improved lower bound has been found). To improve the lower bound further, we would need to find a graph with a huge number of (a,b)-separators, and copy it many times. This procedure is detailed in the paper mentioned earlier (the graph they propose has 144 vertices).