Lecture notes for MATH 363, Wednesday January 20th. Prepared by Nicolas Sonnerat.

1 Graph Theory Basics

We started out reviewing some basic definitions, most of which are illustrated in Figure 1. Recall that the *degree* $d(v)$ of a vertex v is the number of its neighbours, so in the example $d(s) = 3$.

Next, we defined a graph G to be *connected* if there is a path from u to v for any pair of vertices u, v of G. So the example graph given in Figure 1 is not connected, as there is no path from the isolated vertex to s for example.

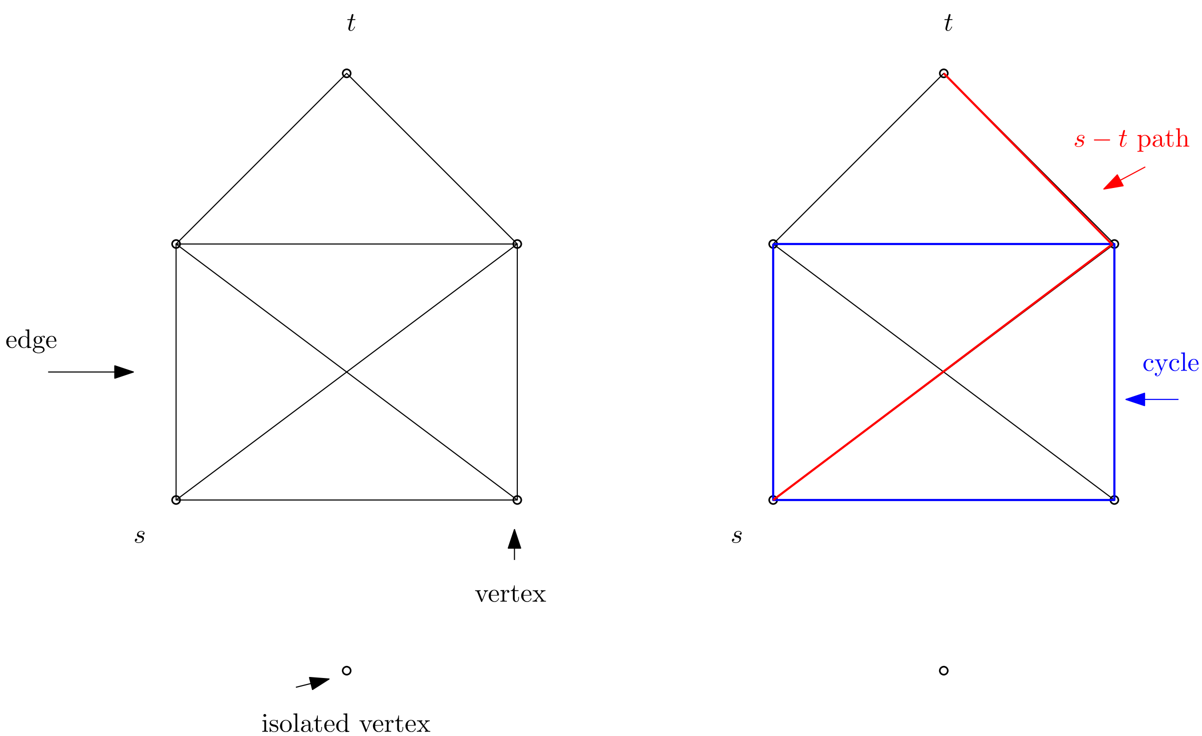

Figure 1: Basics.

The Handshaking Lemma asserts that for any graph,

$$\sum_{v \in V(G)} d(v) = 2|E(G)|$$

which is true because when you sum all the degrees, every edge is counted exactly twice (once for each of its endpoints).

If the edges of a graph are ordered pairs rather than unordered pairs, one obtains a *directed* graph. An example is given in Figure 2.

2 Matchings

Consider the following two motivating examples:

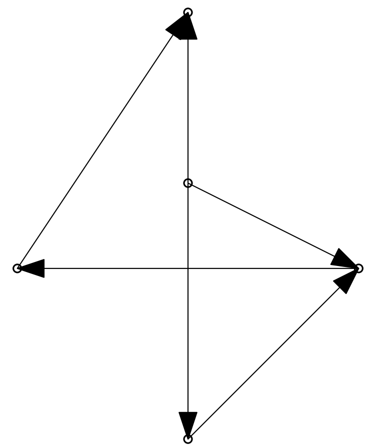

Figure 2: A directed graph.

Example 1 After graduating from medical school, students become resident interns at a hospital. Assigning students to hospitals is a complicated problem, because several things have to be taken into account, such as

- each student has preferences regarding where he/she wants to work
- the hospitals also have preferences as to which students they wish to accept
- different hospitals can take on different numbers of students

and so on. To make the problem more tractable, we make the following simplifying assumptions:

1. each student is either willing to work at a specific hospital or not, i.e. the students' choices are not ranked
2. each hospital can accept exactly one student
3. we ignore the hospitals' preferences, i.e. a hospital will take any student willing to go there.

With these simplifying assumptions, we can model the situation as a graph, where there is a vertex for each student and for each hospital, and an edge between student s and hospital t precisely if student s is willing to go to hospital t (see Figure 3).

Now the goal is to match as many students as possible with hospitals, respecting the students' preferences and the constraint that each hospital can only take on one student.

Before translating the problem into graph theoretic language and solving it, we'll consider another very similar example.

Example 2 Suppose you have a number of jobs waiting to be processed by machines. Not every machine can process any job, so you want to find an assignment of jobs to machines that takes care of the maximum number of jobs in one go. To model this as a graph, take a vertex for each job and each machine, and include an edge (j, m) if job j can be processed by machine m (see again Figure 3). This type of problem comes up all the time in industrial settings, even though in real life things are more complicated than I described them here.

Definition: A *matching* M in a graph G is a set of edges (i.e. $M \subseteq E(G)$) such that every vertex of G is in at most one edge of M.

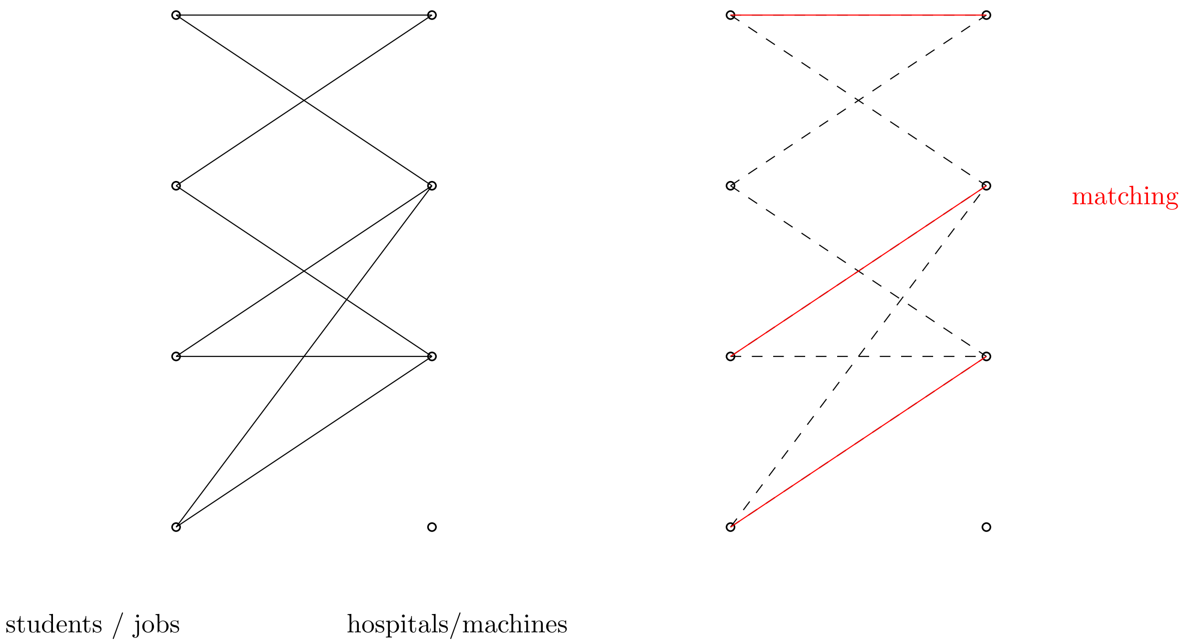

Figure 3: Matchings.

This captures exactly what we're looking for in the two motivating examples: At most one student (job) can be assigned to any hospital (machine), and every student (job) will only be assigned to at most one hospital (job). An example of a matching is shown in Figure 3.

Our goal now is to find a matching of maximum size. The idea behind achieving this is simple but clever. Suppose you have a graph G, a matching M in G, and a path P in G with the following two properties:

1. the two endpoints of P are not in edges of M
2. the edges of P alternate between not being in M and being in M.

Such a path is called an M-*augmenting* path. See Figure 4 for an example. Note that the path is augmenting with respect to a specific matching M; if I take a different matching N, then my M-augmenting path P might not be an N-augmenting path.

Now given such an M-augmenting path P, I can get a matching with one more edge by "switching" the edges of P, i.e. if an edge of P was in M, take it out, if it was not in M, put it in.

This leads to a straightforward solution of the problem of finding a maximum matching:

1. Start with any matching, e.g. $M = \emptyset$ (Exercise: Convince yourself that the empty set is a valid matching.)
2. Find an M-augmenting path P.
3. Switch on P.
4. Repeat steps 2 and 3 until there are no more augmenting paths.

Amazingly, this simple procedure works. Because of the switching procedure, it's clear that if a matching M has an M-augmenting path, then M cannot be maximum (because we can switch and get a bigger matching). What remains to be shown is that if there are no M-augmenting paths, then M is in fact

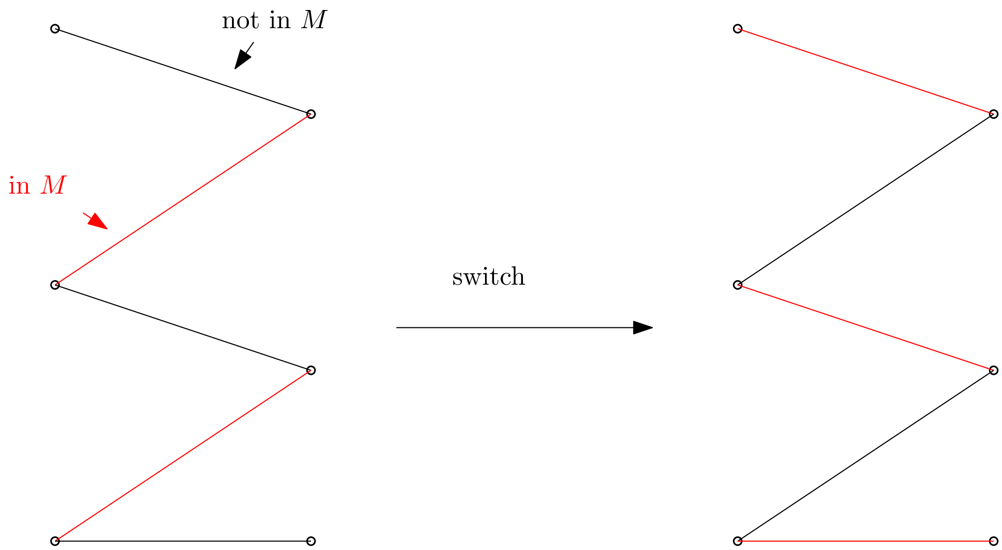

Figure 4: An augmenting path.

maximum. In other words, our straightforward algorithm cannot terminate with a matching for which there are no augmenting paths, but which isn't optimal.

To prove this, we argue by contradiction. Suppose the algorithm found a matching M for which there are no M-augmenting paths, but that this matching is not optimal. This means there must be a different matching N containing more edges (i.e. $|N| > |M|$).

Consider the symmetric difference $N \triangle M$, defined to be the set of edges in EITHER N OR M, but not in both. Each vertex of G is in at most 2 edges of $N \triangle M$, because it is in at most 1 edge of N and at most 1 edge of M. So if we restrict ourselves to the edges of $N \triangle M$, we get a graph consisting of isolated vertices, paths, and cycles (since these are the only graphs where every vertex has degree at most 2). Moreover, the edges of the paths and cycles must alternate between being in M and being in N, because no vertex can be in two edges of M or two edges of N.

This means that the cycles must contain an even number of edges; half from M and half from N, otherwise you get two edges from the same matching sharing a vertex. If a path has an even number of edges, then half must be from M and half from N. But remember that we're assuming that N is strictly bigger than M. This implies that there must be at least one path with an odd number of edges which starts and ends with an edge of N. But this is exactly saying that the path is M-augmenting, and we had assumed that there were no M-augmenting paths, so we've derived a contradiction.

In other words, it cannot be that there is a matching N which is strictly bigger than a matching M for which there are no M-augmenting paths, and the proof is complete.

The question now of course is how we can find these augmenting paths. In general graphs, this is a bit tricky, but in a special class of graphs called bipartite graphs, it's very easy. [The graph in our motivating examples is bipartite.]