COL758: Advanced Algorithms **Spring 2019**

Lecture 14: February 23

Lecturer: Naveen Garg *Scribe: Ayush Verma*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

14.1 Recap

In last class, we have seen

- How to solve shortest path using matrix multiplication
- Saw a $\mathcal{O}(n^w \log n)$ time algorithm for solving APSP for unweighted, undirected graph.

Now we will look at the problem of finding a perfect matching in a bipartite graph.

14.2 Bipartite graph and the perfect matching problem

A graph $G = (V, E)$, is bipartite if the vertex set V can be partitioned into two sets A and B (the bipartition) such that no edge in E has both endpoints in the same set of the bipartition. A matching $M \subseteq E$ is a set of edges such that every vertex of V is incident to at most one edge of M. If a vertex v has no edge of M incident to it then v is said to be unmatched. A matching is perfect if no vertex is unmatched; in other words, a matching is perfect if the cardinality of its bipartition are equal i.e. $|A| = |B|$. The problem is to find a perfect matching if it exists.

Augmenting Path Algorithm

The classic algorithm for finding a perfect matching is the augmenting path algorithm.

Alternating Path

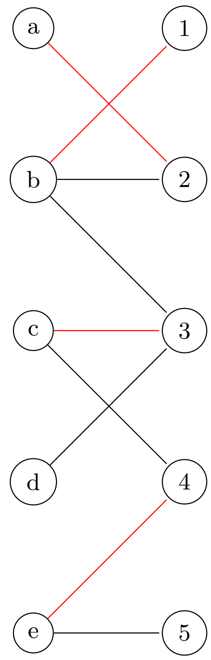
A path in which alternate edges belong to the matching M

Augmenting Path

An alternating path which begins and ends with an unmatched vertex.

Example

Red edge lies in the Matching M and Black edges denotes the other remaining edges.



$a-2-b-1$ is an alternating path, $1-b-3-c$ is an alternating path, unmatched vertices are d and 5, $5-e-4-c-3-d$ is an augmenting path. Now take this augmenting path and flip its edges , i.e. edges of the path in matching are removed and not in matching are put in matching so $5-e-4-c-3-d$ changes to $5-e-4-c-3-d$. Note that the matching has been extended from 2 to 3 in this path and the matching degrees haven't changed hence it's a viable matching. In this we can create a perfect matching if it exists. Now we need to see how to find an augmenting path.

Finding Augmenting Path

We use a DFS-like procedure to get augmenting paths, we start with an unmatched vertex v and go to the matched neighbours of v and then to the unmatched neighbours of that vertex and so on. The DFS tree will be an alternating tree with matched and unmatched vertices at alternating levels or iterations. When you find another unmatched vertex u then we have found an augmenting path v to u , otherwise if we don't find an unmatched vertex other than v in this alternating path then Graph G does not have a perfect matching. Let S denote a subset of the set U that will be included in the augmenting path while T be the subset of V included in the augmenting path , T is also called the neighbourhood of S denoted by $N(S)$, note that $|S|-|T|=1$ as after the starting vertex, for every v in T we include the corresponding u in S. S is called a hall set.

Hall Set

A set $S \subseteq U$ such that $|N(S)|<|S|$ where $N(S)$ is the neighbourhood of S.

Hall Theorem

A bipartite graph $G=(U,V,E)$ has a perfect matching iff it has no hall set.

In the augmenting path finding at every point we find a hall set and flip the matching , increasing the matching by one. A similar algorithm is used for general graphs called the blossom algorithm. The running time for the above classic algorithm is $\mathcal{O}(mn)$ as finding augmenting path is $\mathcal{O}(n)$ and the size of matching increases by one edge every iteration.

14.3 Schwartz-Zippel lemma

If $p(x)$ is an univariate polynomial of degree d we know that P has atmost d roots. Now look at multivariate case. Consider $p(x_1, x_2, ..., x_n)$ to be a non-zero degree d polynomial in n variables over a field F. Let S be a subset of F. Let x_i be picked randomly and independently from S.

$$P[p(x_1, x_2, ..., x_n) = 0] \leqslant \frac{d}{|S|}$$

So this is trivially true for a univariate polynomial, we will prove it for the general case.

$$p(x_1, x_2, ..., x_n) = x_n^k q(x_1, x_2, ..., x_{n-1}) + r(x_1, x_2, ...x_n)$$

therefore using bayes theorem we write:

$$P[p() = 0] = P[p() = 0|q() = 0]P[q = 0] + P[p() = 0|q() \neq 0]P[q() \neq 0]$$

$$P[p() = 0] \leqslant P[q() = 0] + P[p() = 0|q \neq 0]$$

$$P[p() = 0] \leqslant \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}$$

This is because the degree of $q()$ is atmost $d - k$ and when we put $(x_1, x_2, ...x_{n-1})$ in $p()$ we get an atmost k degree polynomial in x_n .
In the next lecture we will use this lemma to get a matrix based method to solve the matching problem in bipartite graphs.