CS601 Algorithms and Complexity **Jul-Nov 2021**

Homework (No submission)

Lecture 19-20 (Sep 27, 28)

- We have n TAs and k courses. For each TA, we are given a list of suitable course. A TA is either a PhD, an Mtech2, or an Mtech1. The requirements of any course are given as one of the following:
  - at least d PhD TAs, at least b Mtech2 TAs, at least c Mtech1 TAs
  - at least d TAs from PhD+Mtech2, at least c Mtech1 TAs
  - at least d TAs from PhD, at least c TAs from Mtech1+Mtech2
  - at least d TAs

  We want to check whether it is possible to fulfill requirements of every course and if it is possible then find a TA allocation. Show that this problem can be converted into an instance of bipartite matching.

  If it is not possible to fulfill all the requirements, then some TAs will have to be asked to expand their list of suitable courses. Which TAs will you ask?

- Let G be a bipartite graph and let the maximum possible size of a matching in G be k (by size we mean the number of edges in a matching). Let's delete every degree 1 vertex along with its neighbor from G, and call the resulting graph as H. Note that when you delete a vertex v, naturally the edges incident on it disappear. Let the maximum possible size of a matching in H be ℓ. Prove that $k = \ell + n_1$, where n_1 is the number of degree 1 vertices in G.

Lecture 21 (Sep 30)

- Let G be a bipartite graph and M be a matching in G. Let us define a new directed graph H, obtained from G as follows: direct all the edges in M from left to right, direct all other edges from right to left, add a source vertex s and a destination vertex t, add edges from s to the free vertices on the right side, and add edges from the free vertices on the left side to t.

  Prove that
  - Any directed path from s to t in H will give us an augmenting path in G with respect to M.
  - Any augmenting path in G with respect to M will give us a directed path from s to t in H.

- Let G be a bipartite graph with n_1 vertices on the left side and n_2 vertices on the right side. Suppose $n_1 - k$ is the maximum possible size of a matching in G. Then prove that there must exist a set S of left side vertices such that the total number of neighbors of S is at most $|S| - k$.

  You can prove this as follows. Consider a matching of size $n_1 - k$. Clearly, there are k free/unmatched vertices on the left side. Since we are at a maximum matching, there should not be any augmenting paths. That means if we start from any of the free vertices on the left side and follow an alternating path (edges alternating between matching and non-matching edges), we will end up on a matched vertex on the left side (why?). It's also possible that the free vertex has no neighbors, in that case, there is simply no alternating path to follow. Now, keeping these in mind, can you construct the set S?

Lecture 22-24 (Oct 4-7)

- In the taxi scheduling problem, we are given a list of taxi bookings and we have to find the minimum number of taxis required to serve all the bookings. We are given a directed graph D, where each vertex represents a booking. An edge from B_1 to B_2 indicates that that there is enough time gap between the two bookings so that it a taxi can first serve B_1 and after that it can serve B_2 as well. No edge between B_1 and B_2 indicates that the same taxi cannot serve both B_1 and B_2. The graph is naturally transitively closed.

  Define a bipartite graph G as follows: for each booking B_i, we have two vertices in G, ℓ_i on the left side and r_i on the right side. For each edge (B_i, B_j) in D, we have an edge (ℓ_i, r_j) in G. Prove that the minimum number of taxis required is same as the number of unmatched left side vertices in a maximum matching in G.