COL 751 - Lecture 1

Course Policy

- Assignments (3 theory, 1 programming): $10 \times 3 = 30\%$
- Exams: $25\% + 30\%$
- Quizzes (2 or more): 10%
- Attendance : 5%
- Research Project (extra weightage): 20%
  Group size : 2

Audit Policy To audit pass one should have 45% in the course total.

Plagiarism Policy Cheating or allowing anyone to copy would lead to a penalty of one grade per assignment / quiz / exam.

Prerequisite

- Basic algorithms (Minimum Spanning Tree, BFS, DFS, shortest path tree for weighted graphs, greedy algorithms, dynamic programming, Max-flow Min-cut theorem).
- Probability theory (linearity of expectation, independent random variables).

1 Course Content

I. Graph Preservers Subgraph preserving certain property of graphs.

Example 1: Sparse graphs preserving connectivity is Spanning Forest.

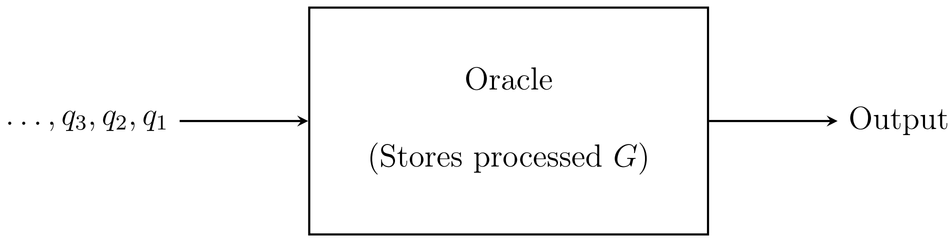
Example 2: Sparse graph preserving distances from a given source vertex is Shortest path tree with $O(n)$ edges.

II. Graph Oracles Data structures for answering certain queries on the graphs.

In oracle model we are suppose to:

- First preprocess the graph to store certain information about the graph in what we call *oracle*.

- Post preprocessing a series of queries are given and we should be able to answer the queries in an efficient way.



Goals: The oracle size should be small, and time to answer queries should be small.

Example 1: What is sparsest oracle to report distances in $O(1)$ time?
Ans: Distance matrix, it takes $O(n^2)$ space.

Example 2: Sparsest oracle to report connectivity in $O(1)$ time?
Ans: We can assign integer labels to all the vertices so that for any $x, y \in V$, label of x and y are identical if and only if they are in same component. This takes $O(n)$ space.

Example 3: Can you have an $O(n)$ sized data structure that given any two vertices x, y in G answers in constant time the max-flow value between x and y.

Main idea: For any graph $G = (V, E)$, you can compute a weighted tree $T = (V, E_T)$ satisfying that

$$min - cut(x, y, G) \;=\; min - cut(x, y, T), \quad \forall x, y \in V.$$

See the example in the figure below.

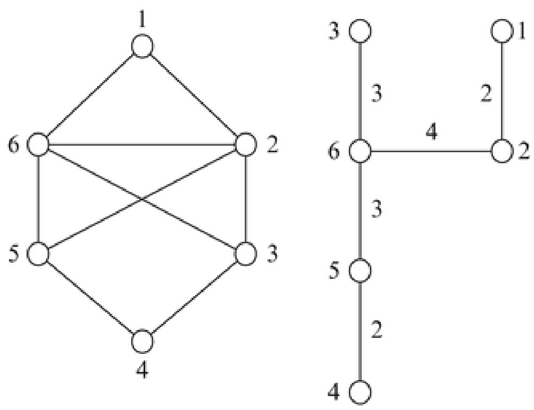


Figure 1: A graph (left), and its Gomory Hu tree (right).

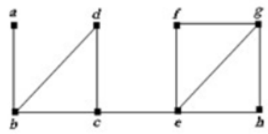
Remark: In subsequent lectures, we will see how to find Gomory-Hu tree, and also how to answer max-flow queries in the Gomory-Hu tree efficiently.

III. Flows, Cuts, Tree Coverings

- Revisit (s, t)-flow and (s, t)-min-cut algorithm.
- Properties of flows and cuts.
- Advanced (s, t)-flow algorithms
- Multi flows, and multi-commodity flows
- Tree coverings: When can we partition edges of graphs into disjoint trees.
- Edmond's Theorem

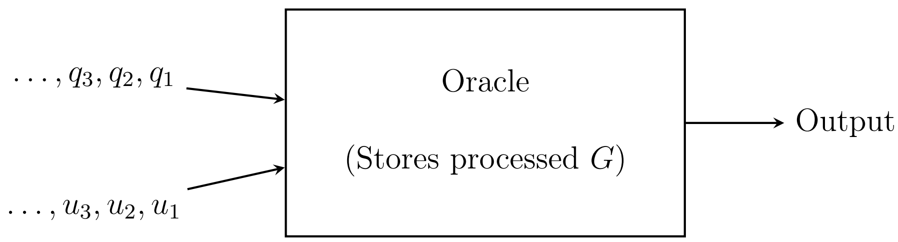
IV. Cut edges, Cut vertices, Edge Connectivity

- How to find cut-edges and cut-vertices efficiently in directed and undirected case.
- Finding edge connected components in directed and undirected graphs.



V. Matching Algorithms

VI. Dynamic Graph Algorithms In the dynamic model at each time stamp and edge is added/deleted from graph.



HW: How can you incrementally maintain MST in $O(\log n)$ amortized time?

Some problems we will look at:

- Can we efficiently maintain MST in fully-dynamic setting?
- Streaming algorithms. Can you maintain this MST efficiently if only $O(\log n)$ space per vertex is allowed?

VII. Miscellaneous Some other domains we will look at (if time permits):

- Algorithms for planar graphs, and planar separator theorem.
- Parameterized algorithms, Tree decomposition and solving problems on graphs with small treewidth.
- Random graphs

2 Important Lemmas

Lemma 1. For any $n \geqslant 2$, we have

$$\frac{1}{4} \leqslant \left(1 - \frac{1}{n}\right)^n \leqslant \frac{1}{e}.$$

Lemma 2. For any n random variables, we have

$$E(X_1 + \cdots + X_n) = E(X_1) + \cdots + E(X_n)$$

Lemma 3. For any n independent random variables and any subset A of outcomes, we have

$$\text{Prob}\big(X_1, \ldots, X_n \in A\big) = \prod_{i=1}^{n} \text{Prob}\big(X_i \in A\big).$$

3 Probability Theory Application

Dominating Set Definition A dominating set for a graph G is a subset D of vertices, such that any vertex of G is either in D, or has a neighbor in D.

Theorem 1 *For any graph $G = (V, E)$ with n vertices satisfying that degree of all vertices is at least d, there exists a dominating set of size at most $1 + \frac{n \log n}{d}$.*

Proof: Let $k = \frac{n \ln n}{d}$.

The existential proof is as follows:

1. Pick a random set $R = \{r_1, \ldots, r_k\}$ of size k such that each r_i is a uniformly random vertex from V and is chosen independent of other vertices in R.

2. For a particular vertex v, the probability that none of the neighbors of v lie in R is

$$\text{Prob}\big(r_1, \ldots, r_k \notin N(v)\big) \;=\; \prod_{i=1}^{k} \text{Prob}\big(r_i \notin N(v)\big).$$

3. Observe that for any v and any $r_i \in R$,

$$\text{Prob}\big(r_i \notin N(v)\big) \;=\; \Big(\frac{n - |N(v)|}{n}\Big)$$

4. So, the probability that v is not dominated by R is upper bounded by

$$\prod_{i=1}^{k} \Big(\frac{n - |N(v)|}{n}\Big) \;\leqslant\; \Big(1 - \frac{d}{n}\Big)^{\frac{n \ln n}{d}} \;\leqslant\; e^{-\ln n} \;=\; \frac{1}{n}.$$

5. Introduce an indicator variable x_v for every vertex v as below:

$$x_v \;=\; \begin{cases} 1 & v \text{ not dominated by } R, \\ 0 & \text{otherwise.} \end{cases}$$

6. Claim: The number of undominated vertices is the sum of indicator variables $\sum_{v \in V} x_v$.

   So, by linearity of expectation, the expected number of undominated vertices is at most $n \cdot \frac{1}{n} = 1$.

7. Hence there exists a set of size $\frac{n \ln n}{d}$ that leaves at most one vertex undominated, add this vertex to the solution.

$\square$

Exercise Given is an $(n + 1)$ length stick with n joints. The stick is dropped from certain height, due to which each joint breaks with probability p independent of other joints. What is the expected number of pieces into which the stick breaks ?