15-451/651 Algorithm Design & Analysis, Spring 2023

Recitation #7

Objectives

- Practice flow
- Practice max flow
- Practice min cost max flow

Recitation Problems

1. **(Capacity Scaling)** The Ford Fulkerson algorithm performs poorly (slowly) when the capacity of augmenting path (i.e. the capacity of the bottleneck) is small relative to the capacities of the other potential augmenting paths in the residual graph. One way to improve this is by ensuring that you always choose an augmenting path with a sufficiently large capacity. Given network graph H, and flow f, we let H_f be the residual graph with respect to flow f, and $H_f(\Delta)$ be the residual graph with all edges of capacity less than Δ removed.

Algorithm 1 Capacity Scaling

```
Initialize Δ ← max_e c(e)                              ▷ Assume this is a power of 2
while Δ ≥ 1 do
    Construct H_f(Δ)
    while ∃s − t path in H_f(Δ) do
        Augment along the s − t path in H_f(Δ)
        Update H_f(Δ) accordingly
    Update Δ ← Δ/2
```

(a) Prove that upon termination, this algorithm will yield the max flow.

(b) Prove that when the outer loop completes completes an iteration for some value Δ and has current flow f, the following inequality holds:

$$|f^*| \leq |f| + m\Delta$$

where $m = |E|$ and f^* is the optimal flow.

(c) Using the above fact, prove that the complexity of the above algorithm is $O(m^2 \log U)$, where U is the maximum capacity over all edges in the input graph.

2. **(Super fast matching)** In lecture, we saw that Dinic's algorithm runs in time $O(n^2 m)$ on any graph, but on some graphs it runs even faster! For instance, in a *unit-capacity* network, where every edge has capacity one, we proved that Dinic's algorithm runs in time $O(m\sqrt{m})$. In this problem we will take this one step further.

   Suppose our graph has one additional restriction (we still keep the unit-capacity restriction): The net flow across every vertex (except s and t) can be at most one. This is equivalent to saying that every vertex other than s and t has either indegree one or outdegree one (but not necessarily both). Such a network is called a "unit network".

   (a) Prove that in a unit network, the number of blocking flows required to find a max flow is at most $O(\sqrt{n})$. (Hint: use a similar argument to the one in lecture for unit-capacity graphs)

   (b) Prove that we can solve the Bipartite Matching problem in $O(m\sqrt{n})$ time.

3. **(Oral homework scheduling)** You've been hired to help the 451 TAs schedule their oral sessions. There are n TAs, and TA i has s_i slots that they need to book a room for. There are m available room bookings, and each TA i has a list L_i of which room bookings $\{1, 2, \ldots, m\}$ would be suitable for them. Since there is a shortage of room bookings, however, the department has started to sell the bookings for money! The j^{th} room booking costs c_j dollars. Your job is to find a way to schedule all of the oral sessions for the minimum amount of money, or report that it is not possible.