Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2024 Prof. R. Wattenhofer

Principles of Distributed Computing
Exercise 13

1 Flow labeling schemes

In this exercise, we focus on flow labeling schemes. Let $G = \langle V, E, w\rangle$ be a weighted undirected graph where, for every edge $e \in E$, the weight $w(e)$ is integral and represents the capacity of the edge. For two vertices $u, v \in V$, the flow between them (in either direction), denoted $\text{flow}(u, v)$, can be defined as follows. Denote by G' the multigraph obtained by replacing each edge e in G with $w(e)$ parallel edges of capacity 1. A set of paths P in G' is edge-disjoint if each edge (with capacity 1) appears in no more than one path $p \in P$. Let $\mathcal{P}_{u,v}$ be the collection of all sets P of edge-disjoint paths in G' between u and v. Then, $\text{flow}(u, v) = \max_{P\in\mathcal{P}_{u,v}} |P|$. As a convention, $\text{flow}(u, u) = \infty$.

Moreover, for two vertices $u, v \in V$, the cut between them, denoted $\text{cut}(u, v)$ is the minimum number of edges that need to be removed from G' to disconnect u from v. The *max-flow min-cut* theorem asserts that $\text{flow}(u, v) = \text{cut}(u, v)$.

Consider the family $\mathcal{G}(n, \hat{\omega})$ of undirected weighted n-vertex graphs with maximum integral capacity $\hat{\omega}$. We will find flow labeling schemes for this family. Given a graph $G = \langle V, E, w\rangle$ in this family and an integer $k \geq 0$, define the relation:

$$R_k = \{(x, y) \in V^2 \mid \text{flow}(x, y) \geq k\}.$$

Question 1 For all $k \geq 0$, show that R_k is an equivalence relation. As a result, R_k partitions V into a collection of equivalence classes $C_k = \{C_k^1, \ldots, C_k^{m_k}\}$, i.e., $C_k^i \cap C_k^j = \emptyset$ (if $i \neq j$) and $\bigcup_i C_k^i = V$. What is the relationship between C_k and C_{k+1}?

According to the solution of Question 1, given G, one can construct a tree T_G corresponding to its equivalence relations. The k'th level of T corresponds to the relation R_k. The tree is truncated at a node once the equivalence class associated with it is a singleton. For every vertex $v \in V$, denote by $t(v)$ the leaf in T_G associated with the singleton set $\{v\}$.

For two nodes x, y in a tree T rooted at r, we define the separation level of x and y, denoted $\text{SepLevel}_T(x, y)$, as the depth of $z = lca(x, y)$, the least common ancestor of x and y; i.e., $\text{SepLevel}_T(x, y) = \text{dist}_T(r, z)$, the distance from z to the root.

Question 2

a) Show that if there exists a labeling scheme for distance for the class of unweighted trees with m nodes $\mathcal{T}_m$ with label size $\mathcal{L}(\text{dist}, \mathcal{T}_m)$, then there is a labeling scheme for separation level for $\mathcal{T}_m$ with label size $\mathcal{L}(\text{SepLevel}, \mathcal{T}_m) \leq \mathcal{L}(\text{dist}, \mathcal{T}_m) + \lceil \log m \rceil$.

b) Recall there is an $O(\log^2 m)$ labeling scheme for distance in unweighted trees of size m. Show that $\mathcal{L}(\text{flow}, \mathcal{G}(n, \hat{\omega})) = O(\log^2(n\hat{\omega}))$.

Question 3 Assume there is an $O(\log^2 m + \log \omega \log m)$ labeling scheme for weighted distance in integer-weighted trees of size m with maximum weight ω (this can be constructed similarly to the one for the unweighted case).

Find a more careful design of the tree T_G which can improve the bound on the label size to $\mathcal{L}(\text{flow}, \mathcal{G}(n, \hat{\omega})) = O(\log n \log \hat{\omega} + \log^2 n)$. **Hint:** Consider the nodes of degree 2 in T_G.

2 Labeling Game

Alice, Bob and Charlie meet for a picnic. Charlie has brought with him 2000 gummy bears. However, he does not like to share his sweets with Alice and Bob. Instead, he proposes the following game. Charlie will draw an unrooted tree T with at most 1000 nodes and assign each node v a unique ID id_v from 1 to 1000; e.g., if the tree has 3 nodes, the IDs might be $2, 10$ and 633, with the other numbers staying unused. He shows the tree to Alice, but not Bob. Alice will then come up with a short bitstring l_v of length at most L for each node v in T and give all these labels to Charlie. Then, Charlie will interact with Bob:[1]

- Charlie chooses a starting node s and gives Bob id_s and l_s.
- Bob always resides at some node v (initially $v = s$). Bob can make requests of the form $req(x)$. If the node u with $id_u = x$ is connected to the node v, Bob will travel to node u, and Charlie provides him with id_u and l_u. However, if the node u is not connected to v, Charlie will eat 1 gummy bear and tell Bob.
- The game ends if either all the gummy bears are gone or Bob has visited each vertex of T at least once (he can visit some vertices multiple times). In the second case, Bob can share the remaining gummy bears with Alice.

Can you help Alice and Bob win some gummy bears from Charlie?

Question 1 Charlie feels very confident that he can eat all the gummy bears. Therefore, Alice is allowed to use very long bitstrings of up to length $L \leq 1000$. Can you come up with a strategy for Alice and Bob to maximize the amount of gummy bears they win?

Question 2 Charlie is not as confident anymore and wants a rematch. Alice now has to come up with shorter bitstrings of length at most $L \leq 20$. However, as a compromise, the tree T now has to be a star graph. Can you come up with a strategy for Alice and Bob to maximize the amount of gummy bears they win?

Question 3 Charlie is a sore loser. Alice again has to come up with bitstrings of length at most $L \leq 20$. However, Charlie can now choose T as an arbitrary tree. Can you come up with a strategy for Alice and Bob to get at least half of Charlie's gummy bears?

Bonus Question 4 Charlie challenges you to one final game. This time, Alice can only use super short bitstrings of length at most $L \leq 10$. T can be an arbitrary tree. However, Alice can specify a starting set S of nodes from which Charlie chooses the first vertex s. The set S must contain at least 2 vertices. Can you come up with a strategy for Alice and Bob to win at least one gummy bear?

[1] The task is inspired from https://rmi.lbi.ro/rmi_2021