Principles of Distributed Computing 05/24, 2017

Exercise 13

Lecturer: Mohsen Ghaffari

1 Random Edge Identifiers

Consider an n-node graph $G = (V, E)$ and suppose that for each edge $e \in E$, we define an $10 \log n$-bit identifier I_e for e by picking each bit at random.

Exercises

(1a) Prove that with high probability, these are unique edge-identifiers. That is, with probability at least $1 - 1/n$, for each two edges $e, e' \in E$ such that $e \neq e'$, we have $I_e \neq I_{e'}$.

(1b) Consider a set $E' \subset E$ of edges with $|E'| \geq 2$. Prove that with probability at least $1-1/n$, there is no edge $e \in E$ such that $\oplus_{e' \in E'} I_{e'} = I_e$. That is, with high probability, the bitwise XOR of the identifiers of this non-singleton edge-set E' is distinguishable from each edge identifier.

2 Graph Sketching for Connectivity

Consider an arbitrary n-node graph $G = (V, E)$, where each node in V knows its own edges. Moreover, we assume that the nodes in V have access to a desirably long string *shared randomness*. Each node should send a packet with size B-bits to the referee, who does not know the graph, so that the referee can determine whether the graph G is connected or not, with high probability. In the class, we saw an algorithm which solves this problem with packet size $B = O(\log^4 n)$. We now improve the bound to $B = O(\log^3 n)$.

Exercises

(2a) Suppose that for each phase of Boruvka's algorithm, instead of having $O(\log n)$ sketches for each node — where each sketch is made of $O(\log^2 n)$ bits, as described in the class — we have just one sketch per node. Show that still, for each connected component, we can get one outgoing edge with probability at least $1/40$.

(2b) Show that $O(\log n)$ phases of the new Boruvka-style algorithm, where per phase we get an outgoing edge from each component with probability at least $1/40$, suffice to determine the connected components, with high probability.

3 Graph Sketching for Testing Bipartiteness

Consider a setting similar to the above problem, where each node v in an arbitrary n-node graph $G = (V, E)$ knows only its own edges. These nodes have access to shared randomness.

Exercise

(3a) Devise an algorithm where each node sends $O(\log^3 n)$ bits to the referee and then the referee can decide whether the given graph $G = (V, E)$ is bipartite or not.

HINT: Think about transforming G into a new graph H such that the number of connected components of H indicates whether G is bipartite or not.