| | |
|---|---|
| **Principles of Distributed Computing** | 05/22, 2019 |
| Exercise 12 | |
| *Lecturer: Mohsen Ghaffari* | |

# 1 Vertex Coloring using All-to-All Communication

Consider an undirected graph $G = (V, E)$ with $n = |V|$ nodes and maximum degree $\Delta = \Omega(\log^3 n)$. Devise a randomized algorithm that with high probability computes a proper coloring of $G$ with $O(\Delta)$ colors in $O(1)$ rounds of all-to-all communication, where in each round each node can send $O(\log n)$ bits to all other nodes.

***Hint:*** *Think about randomly partitioning nodes into several parts, and coloring each part separately, all in parallel.*

***Solution:*** *We first describe a partitioning process, then explain how to use it to obtain a proper coloring of G with $\mathcal{O}(\Delta)$ colors in $\mathcal{O}(1)$ rounds of all-to-all communication, where nodes send $\mathcal{O}(\log n)$ bits to all other nodes.*

*Consider randomly assigning a label $l \in \{1, \ldots, \sqrt{\Delta}\}$ to each vertex (independently, uniformly at random). Define the induced subgraphs $G_1, \ldots, G_l$ where $G_i$ contains vertices labelled $i$ and an edge appears in $G_i$ if both endpoints are labelled $i$. Let $d_G(v)$ and $d_{G_i}(v)$ be the degree of vertex $v$ in graph $G$ and $G_i$ respectively, and $\Delta_i = \max_{v \in V} d_{G_i}(v)$ be the maximum degree of any vertex in $G_i$. By construction, we expect $\frac{n}{\Delta}$ vertices in each $G_i$ (and with high probability at most $\mathcal{O}(\frac{n}{\Delta})$ vertices). Consider a fixed vertex that is assigned label $i$. Since $\Pr[e = \{u, v\} \in G_i] = \frac{1}{\sqrt{\Delta}}$, we see that $\mathbb{E}[d_{G_i}(v) \mid d_G(v) = k] = \frac{k}{\sqrt{\Delta}}$. By Chernoff bounds,*

$$\Pr[d_{G_i}(v) - \frac{k}{\sqrt{\Delta}} > 2\frac{k}{\sqrt{\Delta}}] \leq \exp\left(\frac{2^2(\frac{k}{\sqrt{\Delta}})}{2}\right) = \exp\left(-2\frac{k}{\sqrt{\Delta}}\right)$$

*Since $\Delta \in \Omega(\log^3 n)$ and $k \leq \Delta$, $\Pr[\Delta_i > 2\frac{\Delta}{\sqrt{\Delta}} = 2\sqrt{\Delta}] \leq \exp(-2\sqrt{\Delta}) < \exp(-2\log n) = n^{-2}$. Hence, with high probability, we can color each $G_i$ with $\mathcal{O}(\sqrt{\Delta})$ colors.*

*We now describe how to use the above partitioning:*

1. *Partition $G$ into $G_1, \ldots, G_l$ as described above.*
2. *Assign color sets of size $\mathcal{O}(\sqrt{\Delta})$, to each $G_i$ so that $\mathcal{O}(\Delta)$ colors are used in total.*
3. *Color each $G_i$ with their assigned color sets in $\mathcal{O}(1)$ round.*

*Each vertex can decide locally which $G_i$ it belongs to. Then, vertex $i$ gathers subgraph $G_i$ and performs the $\mathcal{O}(\sqrt{\Delta})$ coloring locally. By our partitioning process, we know that (with high probability) the maximum degree in each $G_i$ is $\mathcal{O}(\sqrt{\Delta})$ and each $G_i$ has $\mathcal{O}(\frac{n}{\sqrt{\Delta}})$ vertices. So, each subgraph $G_i$ can be gathered using $\mathcal{O}(\sqrt{\Delta} \cdot \frac{n}{\sqrt{\Delta}}) = \mathcal{O}(\Delta) \subseteq \mathcal{O}(n)$ messages in total. By Lenzen's routing algorithm, $\mathcal{O}(1)$ rounds of all-to-all communication suffice to gather all the subgraphs.*

# 2 Edge Coloring

Consider an undirected graph $G = (V, E)$ with $n = |V|$ nodes and maximum degree $\Delta$. Each edge $e \in E$ can be colored using several colors, that is, can be assigned a set $C(e) \subseteq \{1, \ldots, q\}$ of colors from a palette of size $q$. A color $c \in C(e)$ is *good* for the edge $e \in E$ if for all neighboring edges $f \in E$ we have $c \notin C(f)$. Suppose that we have access to a *checker* that informs the endpoints $u$ and $v$ of an edge $e = \{u, v\} \in E$ whether the colors in $C(e)$ are good.

(2a) Devise a randomized algorithm that invokes the checker once and with high probability finds a good color for every edge, using a total of $q = O(\Delta \log n)$ colors.

***Solution:*** *Suppose each edge e picks a random color palette*[1] $C(e)$ *of size* $r = 10 \log n$ *out of* $q = 40\Delta \log n$ *colors. A color c is said to be* bad *for an edge e if some neighboring edge picked c. Consider a fixed edge e. Applying union bound over all* $\leq 2\Delta$ *neighboring edges,*

$$\Pr[c \textit{ is bad}] \leq 2\Delta \cdot \frac{r}{q} = 2\Delta \frac{10 \log n}{40\Delta \log n} = \frac{1}{2}$$

*So, the probability that edge e has no good color is at most*

$$\left(\frac{1}{2}\right)^r \leq \exp\left(-\frac{10 \log n}{2}\right) = n^{-5}$$

*Applying union bound over all* $\leq n^2$ *edges in the graph, the probabilty that* some *edge e has no good color in their palette* $C(e)$ *is* $\leq n^2 \cdot n^{-5} = n^{-3}$*. Hence, with high probability, all edges have some good color in* $C(e)$*, and a good color will be found for every edge with one invocation of the checker.*

(2b) Devise a randomized algorithm that invokes the checker twice and with high probability finds a good color for every edge, using a total of $q = O(\Delta \log \log n)$ colors.

***Solution:*** *We use the same process as above, with* $q = \mathcal{O}(\Delta \log \log n)$ *and* $r = \mathcal{O}(\log \log n)$*. One can verify that, for any fixed edge e, all colors in* $C(e)$ *are bad with probability at most* $(\log n)^{-c}$*, for some constant c. We invoke the checker once and assign colors to edges with good colors in their palettes* $C(e)$*. We repeat the same process on the remaining uncolored edges. For each edge, we expect* $\leq 2\Delta \log^{-c} n$ *uncolored neighboring edges, and* $\leq 4\Delta \log^{-c} n$ *uncolored neighboring edges with high probability. One can then show that the probability that all remaining edges have some good color with high probability. Thus, another invocation of the checker will find a good color for all remaining edges.*

(2c) Explain how the above algorithms in (2a) and (2b), respectively, imply algorithms for the routing problem where the maximum number of messages starting from or destined to one node is at most $O(n/\log n)$ and $O(n/\log \log n)$, respectively, in the all-to-all communication setting, where in each round each node can send $O(\log n)$ bits to all other nodes.

***Solution:*** *Given an underlying graph* $G = (V, E)$*, consider a bipartite graph* $H = (S \cup T, E')$ *with* $S = T = V$ *(i.e. H has* $2 \cdot |V| = 2n$ *vertices), and* $\{s, t\} \in E'$ *if there is a message from* $s \in S$ *to* $t \in T$*. Perform the edge coloring algorithms above. Consider an edge* $\{s, t\} \in E'$ *which is assigned color* $c \in \{1, \ldots, q\}$*. To route a message from s to t, we first route the message from s to c, then c to t.*

*If each node has at most* $\mathcal{O}(n/\log n)$ *messages, then* $\Delta = \mathcal{O}(n/\log n)$ *in H. No two adjacent edges are assigned the same color in (2a) and (2b), so every intermediate node (identified by the good edge color) is given at most* $\mathcal{O}(n)$ *messages. Using* $q = \mathcal{O}(\Delta \log n) = \mathcal{O}(n)$ *colors, Lenzen's routing algorithm tells us that all messages can be routed via an intermediate node in* $\mathcal{O}(1)$ *rounds of all-to-all communication. When each node has at most* $\mathcal{O}(n/\log \log n)$ *messages, we use* $q = \mathcal{O}(\Delta \log \log n)$ *colors.*

[1]Consider a random permutation of $q$ colors and picking the first $r$ colors.