Weekplan: Distributed Algorithms III

Philip Bille   Inge Li Gørtz   Christian Wulff-Nilsen

References and Reading

[1] Distributed Algorithms Chapter 6. By Jukka Suomela.

Exercises

1 Larger palette (Ex. 6.1 from [1]) Assume that we have a graph without any isolated nodes. We will design a graph-coloring algorithm $\mathscr{A}$ that is a bit easier to understand and analyze than the algorithm of Section 6.4. In algorithm $\mathscr{A}$, each node u proceeds as follows until it stops:

- Node u picks a color $c(u)$ from $\{1, 2, \ldots, 2d\}$ uniformly at random; here d is the degree of node u.
- Node u compares its value $c(u)$ with the values of all neighbors. If $c(u)$ is different from the values of its neighbors, u outputs $c(u)$ and stops.

Analyze the algorithm, and prove that it finds a 2Δ-coloring in time $O(\log n)$ with high probability.

2 Unique identifiers (Ex. 6.2 from [1]) Design a randomized PN algorithm $\mathscr{A}$ that solves the following problem in $O(1)$ rounds:

- As input, all nodes get value $|V|$.
- Algorithm $\mathscr{A}$ outputs a labeling $f : V \to \{1, 2, \ldots, x\}$ for some $x = |V|^{O(1)}$.
- With high probability, $f(u) \neq f(v)$ for all nodes $u \neq v$.

Analyze your algorithm and prove that it indeed solves the problem correctly.

In essence, algorithm $\mathscr{A}$ demonstrates that we can use randomness to construct unique identifiers, assuming that we have some information on the size of the network. Hence we can take any algorithm $\mathscr{B}$ designed for the LOCAL model, and combine it with algorithm $\mathscr{A}$ to obtain a PN algorithm $\mathscr{B}'$ that solves the same problem as $\mathscr{B}$ (with high probability).

See hint in [1].

3 Large independent sets (Ex. 6.3 from [1]) Design a randomized PN algorithm $\mathscr{A}$ with the following guarantee: in any graph $G = (V, E)$ of maximum degree Δ, algorithm $\mathscr{A}$ outputs an independent set I such that the expected size of I is $|V|/O(\Delta)$. The running time of the algorithm should be $O(1)$. You can assume that all nodes know Δ.

See hint in [1].

4 Max cut problem (Ex. 6.4(a) and (b) from [1]) Let $G = (V, E)$ be a graph. A cut is a function $f : V \to \{0, 1\}$. An edge $\{u, v\} \in E$ is a cut edge in f if $f(u) \neq f(v)$. The size of cut f is the number of cut edges, and a maximum cut is a cut of the largest possible size.

(a) Prove: If $G = (V, E)$ is a bipartite graph, then a maximum cut has $|E|$ edges.

(b) Prove: If $G = (V, E)$ has a cut with $|E|$ edges, then G is bipartite.

5 Max cut algorithm (Ex. 6.5 from [1]) Design a randomized PN algorithm A with the following guarantee: in any graph $G = (V, E)$, algorithm $\mathscr{A}$ outputs a cut f such that the expected size of cut f is at least $|E|/2$. The running time of the algorithm should be $O(1)$. Note that the analysis of algorithm $\mathscr{A}$ also implies that for any graph there exists a cut of size at least $|E|/2$.

See hint in [1].

6 Maximal independent sets (Ex. 6.6 from [1]) Design a randomized PN algorithm that finds a maximal independent set in time $O(\Delta + \log n)$ with high probability.

See hint in [1].

7 Quiz (Section 6.5 of [1]) Consider a cycle with 10 nodes, and label the nodes with a random permutation of the numbers $1, 2, \ldots, 10$ (uniformly at random). A node is a local maximum if its label is larger than the labels of its two neighbors. Let X be the number of local maxima. What is the expected value of X?