CS120: Intro. to Algorithms and their Limitations Hesterberg & Vadhan

Sender–Receiver Exercise 3: Reading for Receivers

Harvard SEAS - Fall 2022 *2022-10-13*

The goals of this exercise are:

- to develop your skills at understanding, distilling, and communicating proofs and the conceptual ideas in them, especially for proofs in graph theory
- to reinforce the definition and algorithms we have seen for Graph Coloring, and introduce the related concept of Independent Sets
- to expose you to a nontrivial exponential-time algorithm

To prepare for this exercise as a receiver, you should try to understand the theorem statement and definition in Section 1 below, and review the material on graph coloring covered in class on October 19. Your partner sender will communicate the proof of Theorem 1.1 to you.

1 The Result

Last time we saw[1] that 2-Coloring can be solved in time $O(n + m)$ via BFS, but for 3-Coloring we have no algorithm but exhaustive search, which can take time $O(m \cdot 3^n)$: there are 3^n ways to pick a color for each of the n vertices, and m edges whose endpoints must be verified to be different colors. Here you will see an algorithm for 3-coloring with a better running time:

Theorem 1.1. *3-Coloring can be solved in time* $O((1.89)^n)$.

Even though this is still exponential, the improvement over 3^n is significant and allows for solving noticeably larger problem sizes. The best known running time for 3-coloring is approximately $O((1.33)^n)$.

A key concept in the proof of this theorem is that of an *independent set*:

Definition 1.2. Let $G = (V, E)$ be a graph. An *independent set* in G is a subset $S \subseteq V$ such that there are no edges entirely in S. That is, $\{u, v\} \in E$ implies that $u \notin S$ or $v \notin S$.

Observe that a proper k-coloring of a graph G is equivalent to a partition of V into k independent sets (each color class should be an independent set).

2 The Proof

Algorithm.

[1] Stated in lecture, proved in detailed lecture notes

Correctness Lemma.

Proof of Lemma.

Runtime.