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

Sender–Receiver Exercise 3: Reading for Senders

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

The idea of the algorithm as follows. Instead of doing exhaustive search for the entire coloring (for which there are 3^n possibilities), we will just do exhaustive search for the smallest color class S, which must be of size at most $n/3$. Once we've fixed a possible choice S for the smallest color class, we just need to check that (a) S is an independent set, and (b) when we remove S, the graph is 2-colorable. Each of these checks can be done in time $O(n+m)$. So our runtime is dominated

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

by the number of sets of size at most $n/3$, which can be shown to be at most c^n for a constant $c < 1.89$.

To justify this reduction to 2-colorability (and checking independence), we prove the following lemma:

Lemma 2.1. *For a graph $G = (V, E)$ and $S \subseteq V$, let $G_{-S} = (V - S, E_{-S})$ where*

$$E_{-S} = \{\{u, v\} \in E : u, v \notin S\}.$$

Then:

1. *If $G = (V, E)$ is 3-colorable, then there is an independent set $S \subseteq V$ of size at most $n/3$ such that G_{-S} is 2-colorable.*

2. *If for some independent set $S \subseteq V$, G_{-S} is 2-colorable, then G is 3-colorable. Moreover, if $f_{-S} : V - S \to \{0, 1\}$ is a 2-coloring of G_{-S}, then a 3-coloring f of G is given by:*

$$f(v) = \begin{cases} f_{-S}(v) & \textit{if } v \notin S \\ 2 & \textit{if } v \in S \end{cases}$$

Proof.

1. Let $f : V \to [3]$ be a proper 3-coloring of G. The three color classes $f^{-1}(0), f^{-1}(1), f^{-1}(2)$ partition V into disjoint independent sets. At least one of these sets must be of size at most $n/3$ (else their union would be of size greater than n). Without loss of generality, let's say $|f^{-1}(2)| \leq n/3$. Let $S = f^{-1}(2)$. Then S is an independent set. Moreover, if we restrict f to $V - S$, it only takes on values 0 and 1, so it gives a 2-coloring of G_{-S}. This is a proper 2-coloring of G_{-S}, since every edge in G_{-S} is an edge of G, and f assigns different colors to the endpoints of every edge of G.

2. Suppose $S \subseteq V$ is an independent set in G, and $f_{-S} : V - S \to \{0, 1\}$ is a 2-coloring of G. Define

$$f(v) = \begin{cases} f_{-S}(v) & \text{if } v \notin S \\ 2 & \text{if } v \in S \end{cases}$$

We will show that f is a proper 3-coloring of G. Let $e = \{u, v\}$ be any edge in G. Since S is an independent set, it is not possible for both endpoints of e to be in S. If exactly one of the endpoints of e is in S, then f will assign one endpoint color 2 and the other endpoint color 0 or color 1 (according to f_{-S}) so e will be properly colored. If both endpoints of e are in $V - S$, then both endpoints are colored according to f_{-S} and hence are properly colored since the edge e is also an edge in G_{-S} and f_{-S} is a proper coloring of G_{-S}.

□

Given this lemma, it follows that the following algorithm is a correct algorithm for 3-coloring.

`3by2Coloring`(G)
Input : A graph $G = (V, E)$
Output : A (proper) 3-coloring f of G, or $\perp$ if none exists
foreach $S \subseteq V$ *of size at most* $n/3$ **do**
  if S *is an independent set in* G **then**
    Construct the graph G_{-S} as defined in Lemma 2.1;
    Let $f_{-S} =$ 2-Coloring(G_{-S});
    if $f_{-S} \neq \perp$ **then**
      Construct a 3-coloring f from f_{-S} as in Lemma 2.1;
      return f
return $\perp$

Algorithm 1: 3-Coloring by reduction to 2-Coloring

For each S, we can can check that S is an independent set and solve 2-coloring on G_{-S} in time $O(n + m)$. Thus, to bound the runtime of Algorithm 1, it suffices to bound the number of subsets of V of size at most $n/3$, which can be shown to be at most c^n for a constant $c < 1.89$, for an overall runtime of

$$O\left((n + m) \cdot c^n\right) \leq O\left(1.89^n\right).$$

(Here we use that $(n + m) = O((1.89/c)^n)$, since $c < 1.89$.)

3 A General Combinatorial Bound

You don't need to cover this during the active learning exercise, but in case you are curious, the following is a useful and quite good asymptotic bound on the number of subsets of $[n]$ of size at most pn for any constant $p \in [0, 1/2]$:

Lemma 3.1. *For $n \in \mathbb{N}$ and $p \in [0, 1/2]$, the number of subsets of $[n]$ of size at most pn is at most c^n for*

$$c = \left(\frac{1}{p}\right)^p \cdot \left(\frac{1}{1-p}\right)^{1-p}.$$

Notice that when $p = 1/2$, we have $c = 2$ (so we get the trivial bound of 2^n), and it can be shown that as p approaches 0, c approaches 1. Plugging in $p = 1/3$ as we needed above, we get

$$c = 3^{1/3} \cdot \left(\frac{3}{2}\right)^{2/3} < 1.89.$$