

To me, being green is everything.

*The Green Giant, on the importance of
St. Patrick's Day to him.*

Problem Party-Scheduling. Suppose you are Irish. You are trying to host a large-scale party in your home in Dublin, Ireland. You have a whole host of activities (dunk-the-brit, drink-a-pint(s), lucky-charms-eating-competition, etc.). You had your attendees fill out a google form about which activities they prefer, and now you must schedule your events. However, your attendees are incredibly churlish, so you would rather cancel the party if even a single attendee has two events they want to go to that are scheduled at the same time.

More formally, suppose there are n people $\{p_1, p_2, \dots, p_n\}$, m time slots, k activities $A = \{a_1, \dots, a_k\}$. Each person p_i is associated with a subset $L_i \subset A$ in which they would like to participate in. The problem is to decide if it is possible schedule all k activities in m time slots such that no person has 2 activities with a conflict, e.g., if person p_i wants to go to activity a_p and a_q , a valid solution should not schedule a_p and a_q in the same time-slot $r \in [m]$. Show that a polynomial time algorithm for this problem would imply a polynomial time algorithm for 3-coloring.

Solution: It is helpful to understand the input of the planning problem we denote as \mathcal{P} :

$\mathcal{P}(P = \{p_1, \dots, p_n\}, A = \{a_1, \dots, a_k\}, L = \{L_1, \dots, L_n\}, m) :$

Decide whether a non-conflicting schedule is possible
given people P , activities A ,
and activity preferences L within m time slots.

We reduce from 3COL as follows:

Make($G = (V, E) : \mathbf{3COL}$) $\rightarrow \mathcal{P}$

Output ($P = [|E|], A = V, L = E, 3$).

Runtime. It is clear that this reduction takes at most polynomial time.

Correctness. We want to show that $G \in \mathbf{3COL} \iff \text{Make}(G) \in \mathcal{P}$.

(\rightarrow) Suppose $G = (V, E) \in \mathbf{3COL}$ and fix some ordering of edges. Then there exists, some coloring $c : V \rightarrow [3]$ such that for all edges $(u, v) \in E, c(u) \neq c(v)$. In $\text{Make}(G)$, we make $|E|$ people, V activities, each edge an activity preference, and 3 time slots. More specifically, the i 'th edge (u, v) correspond to two activities u and v that person i wants to participate in.

Then, we claim coloring c assigns each activity a time slot such that no conflicts occur. First, since c is a function, each activity is assigned a single time slot. Next, for any activity preference (u, v) , $c(u) \neq c(v)$ since c is a valid 3-coloring. Hence, there exists a non-conflicting schedule and $\text{Make}(G) \in \mathcal{P}$ correctly.

(\leftarrow) Note that the implications made in the forward direction are all bi-implications. That is, from a non-conflicting schedule, one may directly construct a valid coloring of G .

To give detail, suppose $\text{Make}(G) = (P = [|E|], A = V, L = E, 3) \in \mathcal{P}$. Then, there exists an assignment $a : A \rightarrow [3]$ such that there does not exist activity preference $(u, v) \in L$ that conflicts. In other words (the contrapositive), $a(u) \neq a(v)$ for all $(u, v) \in L$. Since $A = V$ and $L = E$, this is directly a 3 coloring of $G = (V, E)$. Thus, $G \in 3\text{COL}$.

I am in a very Irish time in my life.

Justin Zhang, March 10, 2026

Problem Independent Set in Irish graphs (Ex. 13.9). Consider a special undirected graph $G = (V, E)$ known as a *Irish* graph, where each set $S \subseteq V$ has at most 20 neighbors outside of S ¹. The goal is to compute the largest independent set in G . Either (a), design and analyze a polynomial time algorithm (faster the better), or (b) prove that a polynomial time algorithm would imply a polynomial time algorithm for SAT.

Hint1: Let I be the largest independent set. What can you say about the size of $|V \setminus I|$?

Hint2: 2^{20} is still a constant.

¹The first independent Irish parliament sat on January 21, 1919, so this is what I am counting as the Irish independence day. If this bothers you, you can replace 20 with 16 for St. Patrick's Day.