

3. (10 points in total) You are asked to create a schedule for the upcoming season of a sports league. Based on records from the previous year, it was decided which pairs of teams should play against each other in the upcoming season. Games are played on Sundays and each team can play at most one game on the same Sunday. The goal is to schedule all games in such a way that the number of weeks is minimized.

This scheduling problem is equivalent to the following problem on graphs (called **EDGE COLORING**): You are given a graph G with vertices t_1, \dots, t_n corresponding to the teams. The edges of the graph correspond to the set of games that are to be scheduled in the upcoming season. An *edge coloring* of G assigns to each edge (t_i, t_j) in the graph a “color” w_k such that no vertex is incident to more than one edge of the same color. (This constraint ensured that all games (t_i, t_j) with the same color w_k can be held in the same week.) Your goal is to find an edge coloring of G that uses as few colors as possible.

Give an efficient algorithm to compute an edge coloring of G that uses at most $2d - 1$ different colors, where d is the maximum degree in the graph. Show that the approximation ratio of this algorithm is at most 2.

Solution

Algorithm

Algorithm 1

```

1:  $C \leftarrow 0$                                 ▷ This represents the maximum color id used to color any edge
2: while  $\exists$  uncolored edge  $e$  do
3:    $u, v \leftarrow$  end point  $e$ 
4:    $c_e \leftarrow$  minimum color from 1 to  $C+1$  such that no edge incident from node  $u$  or  $v$  has that color
5:    $\text{color}(e) \leftarrow c_e$ 
6:    $C \leftarrow \max\{C, c_e\}$ 

```

Proof of Correctness

Lemma 1. *The value C never exceeds $2d-1$ where d is the degree of the graph*

Proof. First note that c_e always gets a value at step 4 of the algorithm. This is because all of the remaining edges are using values at most C . So worst case we can always assign the value $C+1$ to c_e .

The color of an edge only depends upon the color of the edges incident on its end points. Consider the case when there are $d-1$ edges incident on u (excluding e) and there are $d-1$ edges incident on v (excluding e) where u, v are the end points of e . Call these sets E_u and E_v . Note that the maximum cardinality from both these sets can be $d-1$ since the maximum degree of the graph is d .

Since the value that we assign to c_e is the minimum value not present in the union of set E_u and E_v , the worst case which will force the value of c_e picked to be maximum is when both E_u and E_v have cardinality $d-1$ and both sets are completely disjoint. In this case the cardinality of the union of these sets would be $2d-2$. So in this case e cannot have any of those $2d-2$ colors. In this case we can assign the

$(2d - 1)^{th}$ color to e . If the cardinality of either of them is less than $d-1$ we can always find a color between 1 to $2d-1$. Or if there is an overlap between sets, then also we can find a color between 1 and $2d-1$.

This means that in every iteration we can find a value to color the edge which is at most $2d-1$. This completes the proof. \square

So this lemma proves that this algorithm takes at most $2d-1$ colors. Also since this algorithm terminates only when all edges have been colored, this algorithm computes a valid edge coloring of the graph.

Analysis

The running time of this algorithm can be bounded by the fact that we find a color in max $2d-1$ steps for each edge e . So running time = $O(E * (2d - 1))$

Proving the approximation bound

Note that any valid edge coloring would use at least d colors where d is the maximum degree of the graph. This is because a node would have d edges coming out of it, all of which would have a different color. Let OPT denote the optimal algorithm to solve the edge coloring problem. Let the greedy algorithm above be called as G . This means

$$OPT \geq d.$$

Also

$$G \leq 2d-1$$

$$G \leq 2d$$

$$G \leq 2 * OPT$$

$G/OPT \leq 2$, which shows that the approximation ratio of the greedy algorithm is at most 2.