

In-Class Activity #3

Vancouver Summer Program

Algorithms

(Tolerant colourings.) Consider a graph where every vertex has degree at most 25. We would like to colour the graph using two colours, *red* and *black*, such that no vertex has more than 12 adjacent vertices of the same colour as itself. To achieve this, your friend Mercator suggests the following scheme: For each vertex in the graph, toss a coin and if the coin toss results in a head then colour that vertex red otherwise colour it black. Label each vertex that has more than 12 adjacent vertices of the same colour with a *FixMe* label. Pick any vertex that has a *FixMe* label and call this vertex v . (If no such vertex exists then stop.) If v is coloured red then recolour it black; if v is coloured black then recolour it red. Update all the *FixMe* labels. Repeat the recolouring process until there is no vertex with a *FixMe* label.

Question: Show not only that Mercator's algorithm will stop but that it will do so in no more than $\Theta(m)$ colouring/recolouring and label update *iterations* where m is the number of edges in the graph.

(It is possible to show that the entire algorithm can run in $\Theta(m + n)$ worst-case time where n is the number of vertices but you do not have to include such a proof in your solution.)