Graph Algorithms : Home Assignment

Rules Every result from the course and the exercise sessions may be used by making an explicit referrence to it (e.g. "Every graph G is $(\Delta(G)+1)$-colourable [course, Chapter 2]."). Brainstorming between students is allowed, but the redaction in each assignment must be distinct.

For information, I have already graded 2 assignments:

- ChatGPT 3.5 scored $2/20$.
- ChatGPT 4 scored $7/20$.

Your goal is to perform strictly better than these AI. Only grades above $10/20$ will be taken into consideration for the final grade.

Exact exponential algorithms for the Graph Colouring Problem

The goal of this exercise is to solve the colouring problem exactly for any graph, with an algorithm of exponential complexity as small as possible. For an exponential complexity, we rely on the notation $O^*(\cdot)$, that satisfies $n^{O(1)}c^n = O^*(c^n)$ (we ignore any polynomial factor).

Given an integer $k \geq 3$, to solve the problem k-COLOUR on an n-vertex graph G, we can enumerate all mappings $\phi\colon V(G) \to [k]$, and test if any of them is a proper k-colouring of G. This yields an algorithm of complexity $O^*(k^n)$. We will see that it is possible to do (much) better.

1. To begin, let us find a first non-trivial algorithm for 3-COLOUR.

   (a) Let T be an n-vertex tree. Show that the number of proper 3-colourings of T is $3 \times 2^{n-1}$. 2 pts

   (b) Use the previous fact to derive an algorithm that solves 3-COLOUR in time $O^*(2^n)$. 2 pts
   Hint: Every connected graph has a spanning tree.

2. Let us now find an even better algorithm to solve 3-COLOUR. Given a graph G, a *dominating set* of G is a set of vertices $X \subseteq V(G)$ such that $N_G[X] = V(G)$ (every vertex $v \in V(G) \setminus X$ has a neighbour in X).

   (a) Let X be a dominating set of G, and $\phi\colon X \to [3]$ a proper 3-coloring of $G[X]$. Show that it is possible to test in polynomial time whether ϕ extends to a proper 3-colouring c of G (we must have $c(x) = \phi(x)$ for every $x \in X$). 4 pts
   Hint: You may find a polynomial reduction to 2-SAT, which is known to be in P.

   (b) Assume that G is connected and that $n \geq 2$. Let T be a BFS-tree of G. Show that the layers of T may be distributed into 2 disjoint dominating sets of G. Deduce an upper bound (as a function of n) for the size of a smallest dominating set of G. 3 pts

   (c) Derive from the above an algorithm that solves 3-COLOUR in time $O^*((\sqrt{3})^n)$. 2 pts

3. Let us now attack the general k-COLOUR problem. We will use a dynamic programming approach.

   (a) Given a graph G, assume that you have already computed all subsets of vertices $X \subseteq V(G)$ such that $\chi(G[X]) \leq j$, for some integer $1 \leq j < k$. Show that you can compute all subsets of vertices $Y \subseteq V(G)$ such that $\chi(G[Y]) \leq j+1$ in time 4 pts

   $$\sum_{t=0}^{n} \binom{n}{t} O^*(2^t) = O^*(3^n).$$

   (b) Derive an algorithm that solves k-COLOUR in time $O^*(3^n)$. What is its space complexity? 3 pts