Let $G = (V, E)$ be an undirected and unweighted graph. A *colouring* of G is an assignment of vertices to colours such that no pair of adjacent vertices share the same colour. This problem is in NP-C; however, we will come up with an exponential-time dynamic programming. We consider the k-Colouring problem, described below.

Instance. An undirected and unweighted graph $G = (V, E)$.

Task. *Is there a colouring of G using at most k colours?*

(a) Describe a brute-force algorithm for k-Colouring. What is the running time of such an algorithm?

(b) An *independent set* of G is a subset of vertices $S \subseteq V$ such that no pair of vertices in S are adjacent. How do independent sets of G relate to colour classes of G?

(c) Prove that there exists an optimal k-colouring such that a colour class is a maximal independent set.

We are now ready to describe an "efficient" algorithm.

(d) We enumerate over all subsets of V. For a subset $S \subseteq V$ of vertices, we define $G[S]$ to be the graph of G *induced* by the vertex set S.

(i) Let $\textsc{OptColour}(S)$ denote the minimum k such that $G[S]$ is k-colourable. Explain why

$$\textsc{OptColour}(S) = 1 + \min\{\textsc{OptColour}(S \setminus I) : I \text{ is a maximal indep. set in } G[S]\}.$$

(ii) What is a suitable base case for this problem?

(iii) What is the final solution, and what is the order of computation?

(iv) On a graph with n vertices, assume that all maximal independent sets can be generated in time $O^*(3^{n/3})$. Here, $O^*(\cdot)$ omits all polynomial factors; that is, $O^*(a^n) = O(p(n) \cdot a^n)$ where $p(n)$ is a polynomial in n.

Show that $\textsc{OptColour}(S)$ can be solved in time $O^*\left(3^{|S|/3}\right)$.

(v) Show that the running time is given by $O^*\left((1 + 3^{1/3})^n\right)$. This gives an approximately $O^*(2.4423^n)$ algorithm.

Hint. *The binomial theorem might come in handy.*

Note. *This is the algorithm described in [Lawler, '76]. This was the best known algorithm until a new inclusion-exclusion algorithm was introduced in 2006 by Björklund and Husfeldt that runs in $O^*(2^n)$; this is the currently best-known algorithm.*

Solution.

(a) There are $k^{|V|}$ possible ways to assign k colours to $|V|$ vertices, so we can generate each possible assignment and verify if it is valid by checking that each edge does not share a similarly colour vertex on each side. This will take $O(k^{|V|} \cdot |E|)$.

(b) Independent sets of G represent potential colour allocations. Since no two vertices are adjacent, they can each be safely coloured the same colour.

(c) Start by selecting a maximal independent set M in G and colour each of them the same. We can then remove each vertex in M from G and recursively find the next maximal independent set. Since each vertex is set a maximal and independent at the time of colouring, no further vertices can be added. Each colour class is a maximal and independent set within the context of the vertices remaining at each step. Therefore, at least one (and possible all) of the colour classes in an optimal k-colouring is a maximal independent set.

(d) (i) This algorithm describes the process from part (c), where we find the maximal independent subset, remove the coloured vertices and repeat the proecss, adding 1 at each layer. This is optimal since the colourings chosen are based on the maximal possible selections of vertices and hence cannot be improved.

(ii) We can define a simple and obvious base case of $\textsc{OptColour}(\varnothing) = 0$.

(iii) Our final solution will simply be $\textsc{OptColour}(V)$.

(iv) In the worst case, we will have a colouring of $k = n$ separate colours. We can calculate the time complexity of this as our worst case, being $\displaystyle\sum_{i=1}^{n} O^*(3^{i/3})$. Before we handle the sum, we must prove that $O^*(n) + O^*(m) = O^*(n+m)$. We'll let $f(n) = c_1 \cdot n^{k_1} \cdot g(n)$ and $h(m) = c_2 \cdot m^{k_2} \cdot p(m)$, where $g(n)$ and $p(m)$ dominate the growth.

- If $m \approx n$, then we will take:
$$f(n) + h(m) \approx (c_1 n^{k_1} + c_2 n^{k_2}) \cdot 2^n,$$
and so in this case, $O^*(n) + O^*(m) = O^*(n+m)$.
- If n and m are not necessarily equal, then, without loss of generality, take n to be the larger one. 2^m is then bounded by 2^n and thus the dominant term still grows faster than some exponential function in $n+m$, so $f(n)+f(m)$ can be bounded by a function of the form $c(n+m)^k \cdot 2^{n+m}$.

Since we know $O^*(n) + O^*(m) = O^*(n+m)$, we can finish the proof by simplifying it as a geometric series:

$$\begin{aligned}\sum_{i=1}^{n} O^*(3^{i/3}) &= O^*\left(\sum_{i=1}^{|S|} 3^{i/3}\right)\\ &= O^*\left(\frac{3^{1/3}}{3^{1/3}-1}(3^{|S|/3}-1)\right)\\ &= O^*\left(3^{|S|/3}\right).\end{aligned}$$

(v) Recall that the complexity for each subset S of vertices in terms of generating all maximal independent sets is $O^*(3^{|S|/3})$. The total complexity of solving the problem over all subsets S can be expressed by considering all sizes s of the subsets and applying the complexity for each size. That is,

$$\sum_{s=0}^{n} \binom{n}{k} O^*(3^{k/3}) = O^*\left(\sum_{s=0}^{n} \binom{n}{k} 3^{k/3}\right).$$

By the binomial theorem, where $(x+y)^n = \displaystyle\sum_{s=1}^{n} \binom{n}{s} x^{n-s} y^s$, we see that this expression simplifies to $O^*\left((1+3^{1/3})^n\right)$.