6. 10 marks **(Take Home Question.)** For a graph G, a function $c : V(G) \to \{1, 2, \cdots \ell\}$ is a *proper coloring of G using at most ℓ colors* if for each $\{u, v\} \in E(G)$, $c(u) \neq c(v)$.

In the LARGE COLORING problem we are given a graph G on n vertices and an integer k, and the goal is to check if G has a proper coloring using at most $n-k$ colors. Design a kernel for LARGE COLORING.

For any graph G, let $\bar{G}$ denote complement graph: graph with the same set of vertices, and edge set $\{\{u,v\} : u, v \in V(G),\ uv \notin E(G)\}$

RR0: If G is empty, if $k \le n$ return True, else return False.
If G contains only isolated vertices, if $k \le n-1$ return True, else Return False.

RR1: If RR0 is not applicable, if an instance (G, k), G has isolated vertex v return $(G - \{v\}, k-1)$

RR2: If RR0 & RR1 are not applicable and G has $\geq 3k+1$ vertices, perform crown decomposition of $\boxed{\bar{G}}$ with the parameter k.
a) If we get a matching M : $|M| \geq k+1$, return "Yes" instance
b) Else we have a crown decomposition (C, H, R), for Crown C & Head H.
Return $(G \setminus (C \cup H), k - |H|)$.

Note: We can perform RR2 always by Crown Lemma.

Safeness of RR0:

G is empty, we can color it with 0 colors, i.e. $k \le n$.
However, if $k > n$ we cannot do it. So, this rule is safe.

If G has only isolated vertices, we need exactly one color to color all the vertices. However, we cannot do it with 0 colors. Thus, we need $k \le n-1$ for it to be Yes instance.

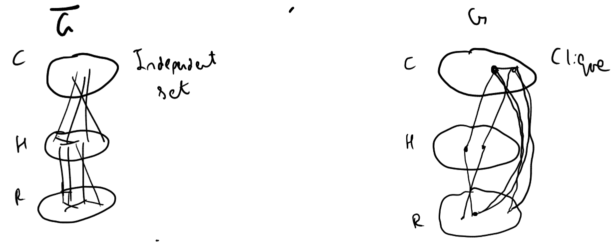
Safeness of RR1

If G has isolated vertices, we can arbitrarily assign it the color "1". Now since RR0 is not applicable, G has at least one edge, & we need at least 2 colors to color the graph. Thus, the color "1" can be used to color the isolated vertex v.

Safeness of RR2 :

If we get a matching of size $k+1$ in $\bar{G}$, we can color pairs of matched vertices the same color. Note: The vertices in a pair have same color, the pairs themselves have different color. Also, assign each unmatched vertex its own color. Thus, we get a coloring with at most $n-(k+1)$ colors. This is proper because vertices in a matching in $\bar{G}$ do not have a connecting edge in G.

If the Crown Lemma returns a Crown Decomposition instead consider below:



Yes instance in (G, k) => Yes instance in $(G \setminus (C \cup H), k - |H|)$

Let $c : V(G) \to \{1, 2, 3 \ldots\ldots |V(G)| - k\}$ be a proper coloring of G.

Since C forms ind. set in $\bar{G}$, C is a clique in G. Also, since C & R have no edges in-between in $\bar{G}$, every vertex in C has an edge to every vertex in R.

Now, for some vertex $v \in C$, $c(v)$ cannot be used for any other vertex in C or R. Thus, all the colors used for vertices in C are distinct and different from the ones in R.

Since vertices in $V(G)$ can be coloured with $n-k$ colors, and the above observation, vertices in R can be colored with at most $|R|+|H|-k$ colors. Now relabel colors in C such that colors in R belong to the set $\{1, 2, 3 \ldots |R|+|H|-k\}$, and vertices in C are colored with vertices in $\{|R|+|H|-k+1, |R|+|H|-k+2, \ldots |R|+|H|+|C|-k\}$.

Consider the coloring $c_1 : R \to \{1, 2, 3 \ldots\ldots |R|+|H|-k\}$, such that $c_1(r) = c(r)$ $\forall\, r \in R$. Since $G \setminus (C \cup H)$ is a subgraph of G, the coloring c_1 is proper for $G \setminus (C \cup H)$.

Yes instance in (G,k) $\Leftarrow$ Yes instance in $(G \setminus (C \cup H), k-|H|) \equiv G'$

Consider some proper coloring $c_1 : R \to \{1,2,3 \ldots, |R|-k+|H|\}$ which colors vertices in the reduced instance G'.

Now, we know $\exists$ a matching which saturates H in the original instance. For the original instance, consider the full-coloring from $V(G) \to \{1,2,3 \ldots |H|+|C|+|R|-k\}$

→ For $r \in R$, same color as $c_1(r)$

→ For vertices in C, color each with distinct colors in the set $\{|H|+|R|-k+1, |H|+|R|-k+2, \ldots |H|+|R|+|C|-k\}$

→ For vertices in H, color with color of matched partner in the matching which saturates H.

<u>To prove C is proper:</u>

Consider 6 possible vertex sets for an edge $\{a,b\}$

a	b	Reason for distinct color
R	R	→ Reduced graph is Already a "Yes" instance
C	C	→ Each color is distinct among $|R|-k+|H|+1$ to $|R|-k+|H|+|C|$
H	H	→ Since these vertices take up color of matched vertices in C, each color of H is distinct
R	H	→ R has color in $\{1,2,\ldots,|R|-k+|H|\}$, H has color in $\{|R|-k+|H|+1, |R|-k+|H|+|C|\}$
C	H	→ Matching in $\bar{G}$ implies no edge in G, so all end-pts. have distinct colors
C	R	→ R has color in $\{1,2,\ldots,|R|-k+|H|\}$, C has color in $\{|R|-k+|H|+1, |R|-k+|H|+|C|\}$

Thus, the coloring described above is proper

Thus, RR2 is also safe.

Theorem : RR0, RR1 & RR2 form a kernel for the problem LARGE KERNEL.

Proof : We apply RR0, RR1 until we cannot do anymore. Thus, there are no isolated vertices. Since we apply RR2 exhaustively, by the time RR2 is done, we have $\leq 3k$ vertices. Also, we have $\leq \binom{3k}{2}$ edges. Thus, the # of vertices in bounded by the computable fn. $f(k) = 3k$. So, it is a valid kernel with # of vertices bounded by $3k$.

7. **(Take Home Question.)** Consider a parameterized problem $\Pi \subseteq \{0,1\}^* \times \mathbb{N}$. Suppose there is a fixed constant c and an algorithm $\mathcal{A}$, that given $(x,k) \in \{0,1\}^* \times \mathbb{N}$, in time bounded by $(\log n)^k \cdot n^c$, correctly determines whether or not $(x,k) \in \Pi$; here n is the number of bits required to represent the pair (x,k). Answer the following questions and in each case prove why your answer is correct:

(a) $\boxed{\text{1 mark}}$ Is $\mathcal{A}$ an XP-algorithm for Π?

(b) $\boxed{\text{9 marks}}$ Is $\mathcal{A}$ an FPT-algorithm for Π?

a) <u>Definition of XP</u>: A problem $L \subseteq \{0,1\}^* \times \mathbb{N}$ is said to be XP if $\exists$ an algo. A s.t. A correctly decides if an instance $(x,k) \in \{0,1\}^* \times \mathbb{N}$ belongs to L or not in time bounded by $f(k) \cdot |(x,k)|^{g(k)}$ for computable fns. $f: \mathbb{N} \to \mathbb{N}$ & $g: \mathbb{N} \to \mathbb{N}$.

Note that $\log n \leq n \quad \forall\, n \in \mathbb{N}$

Also, $\boxed{n = |(x,k)|}$ for some $(x,k) \in \{0,1\}^* \times \mathbb{N}$

$\Rightarrow$ A runs in time bounded by $(\log n)^k \cdot n^c \leq n^k \cdot n^c = n^{(k+c)}$

So, A runs in time bounded by $f(k) \cdot n^{g(k)}$ for $f(k) = 1$
$g(k) = k + c$
for some constant c.

Clearly, f & g are computable.
Thus, A is an XP-algorithm

b)

<u>Lemma</u>: $(\log n)^k \leq 2^{k^2}\, n$ for integer $k \in \mathbb{N}$

$\log n = 2^{\log \log n}$ [Note: Assume all "log" are "$\log_2$"]

$(\log n)^k = 2^{k(\log \log n)}$

Note: $(\log \log n - k)^2 \geq 0$ [Square is non-negative]

$\Rightarrow k \log \log n \leq \dfrac{k^2 + (\log \log n)^2}{2} \leq k^2 + (\log(\log n))^2$

$$(\log n)^k = 2^{k \log\log n} \leq 2^{k^2 + (\log\log n)^2}$$
$$= 2^{k^2} \cdot 2^{(\log\log n)^2}$$

<u>Note</u> : $2^{(\log\log n)^2} \leq n$

<u>Reason</u> : The $2^{(\log\log x)^2} - x$ is a decreasing fn. for $x > 1$ & is negative for $x > 1$.

$$(\log n)^k \leq 2^{k^2}\, 2^{(\log\log n)^2} \leq 2^{k^2} \cdot n$$

Lemma proved.

We know a problem $L \in \{0,1\}^* \times \mathbb{N}$ is FPT iff $\exists$ an algo A s.t. A decides correctly if $(x,k) \in \{0,1\}^* \times \mathbb{N}$ $\in L$ or not in time bounded by $f(k) \cdot |(x,k)|^c$ for some constant C & computable function $f : \mathbb{N} \to \mathbb{N}$

A runs in time bounded by $(\log n)^k \cdot n^c \leq 2^{k^2} \cdot n^{(c+1)} \leq \lceil 2^{k^2} \rceil n^{c+1}$

Clearly, $f(k) = \lceil 2^{k^2} \rceil$ is from $\mathbb{N} \to \mathbb{N}$, & is also computable.

For $c' = c+1$, f as above, the algo A is FPT for the problem π.