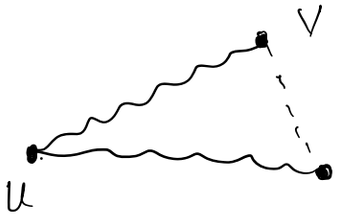
Last Time: Local Search Algorithm for finding min degree spanning tree

Today: Min Degree Spanning Tree (contd.)
Edge Coloring Algorithm

Algorithm

1. Initialize to arbitrary spanning tree T

2. **while** $\exists$



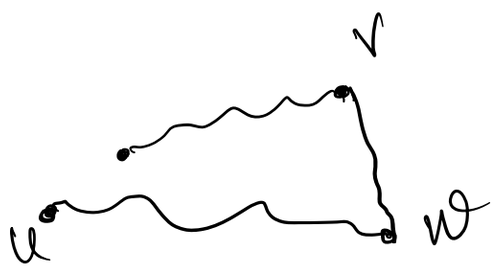
$d_T(v) \leq d_T(w) - 2$

$d_T(w) \leq d_T(u) - 2$

$(v,w) \in E \setminus T$

$d_T(u) \geq \underbrace{\Delta(T) - \log_2 n}$

do



L1: Algo outputs a tree such that
max degree $\leq 2\,OPT + \log_2 n$

L2: Algo runs in poly. time.

Proof:

Potential method: to measure progress in each step.

For a tree T

$$\phi(T) = \sum_{v \in V} 3^{d_T(v)}$$

How much does potential change in one iteration?

Max. possible potential $\leq n \cdot 3^n$

$\rightarrow$ Ham path

Least possible potential $\geq (n-2) \cdot 3^2 + 2 \cdot 3 > n$

$i \geq \Delta(T) - \log_2 n$

$\deg_T(u)$ $\quad i \rightarrow i-1$

$\deg_T(w)$, $\deg_T(v)$ $\quad d_w \rightarrow d_w + 1$, $d_v \rightarrow d_v + 1$

Overall decrease in potential function

$$\phi(T') = 3^{i-1} + 3^{d_w+1} + 3^{d_v+1} + \phi(T) - 3^{i} - 3^{d_w} - 3^{d_v}$$

$$\phi(T) - \phi(T') = 3^{i-1} \cdot 2 + \left(3^{d_w}(-2) + 3^{d_v}(-2)\right)$$

$$\geq 2 \cdot 3^{i-1} - 4 \cdot 3^{i-2}$$

$$= 3^{i-2}(2) = 2 \cdot 3^{i-2}$$

$$\geq 2 \cdot 3^{\Delta(T) - \log_2 n - 2}$$

$$= \Omega\left(\frac{3^{\Delta(T)}}{n^2}\right) \cong \frac{1}{n^3}\phi(T)$$

$$n \cdot 3^{\Delta(T)} \geq \phi(T)$$

$$\Rightarrow 3^{\Delta(T)} = \frac{\phi(T)}{n}$$

$$\phi(T) - \phi(T') \geq \frac{1}{n^3}\phi(T)$$

$$\phi(T') \leq \phi(T)\cdot\left(1-\frac{1}{n^3}\right)$$

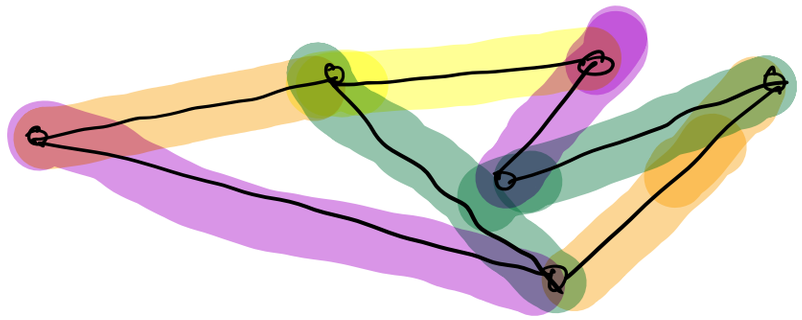
We want to find the smallest r s.t.

$1-x \leq e^{-x}$

$$n3^n \cdot \left(1-\frac{1}{n^3}\right)^r < n \qquad \left| \; e^{r/n^3} > 3^n \right.$$

$$n\cdot 3^n \cdot e^{-r/n^3} < n\,1 \qquad \left| \; r = O(n^4) \right.$$

Edge Coloring

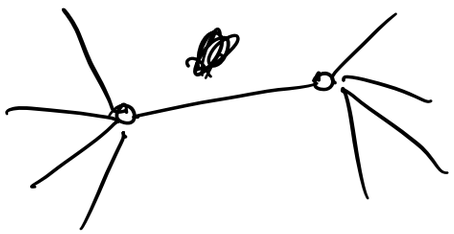


* Assign colors to the edges of a graph such that no two edges that share an endpoint get the same color.

Theorem: poly-time algo. to color using $\Delta + 1$ colors

Δ = max-degree of the graph

Hardness result: In general, NP-complete to decide whether Δ colors are sufficient.

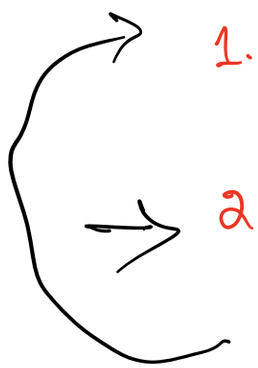


each edge has $\leq 2(\Delta-1)$ nbrs edges

palette of $\Delta+1$ colors

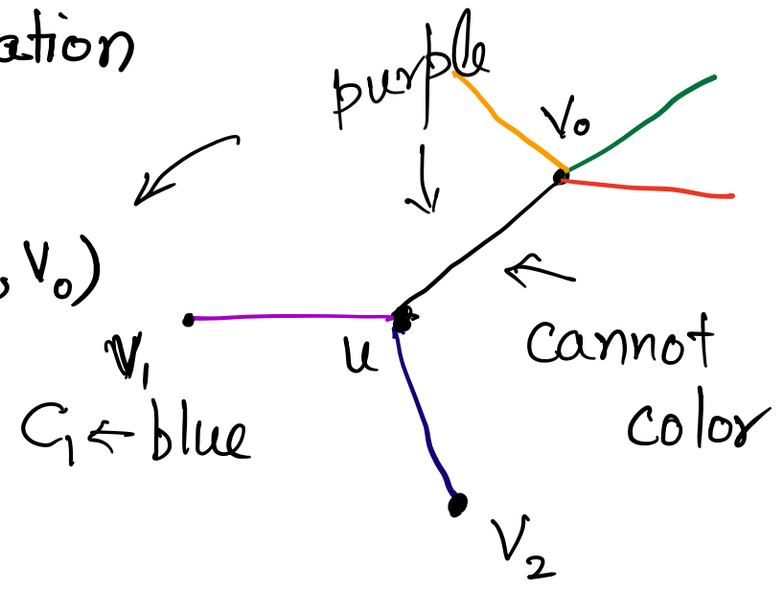
input: $G=(V, E)$ of max. degree Δ

Algorithm: (High-level)



1. Color greedily with colors from palette.
2. If no color is available for an edge, we make some "local modifications"

Algorithm (Detailed) - single iteration



1. Pick uncolored edge (u, v_0)

$i = 0$

2. repeat :

   if $\exists$ color c that both v_i & u lack :

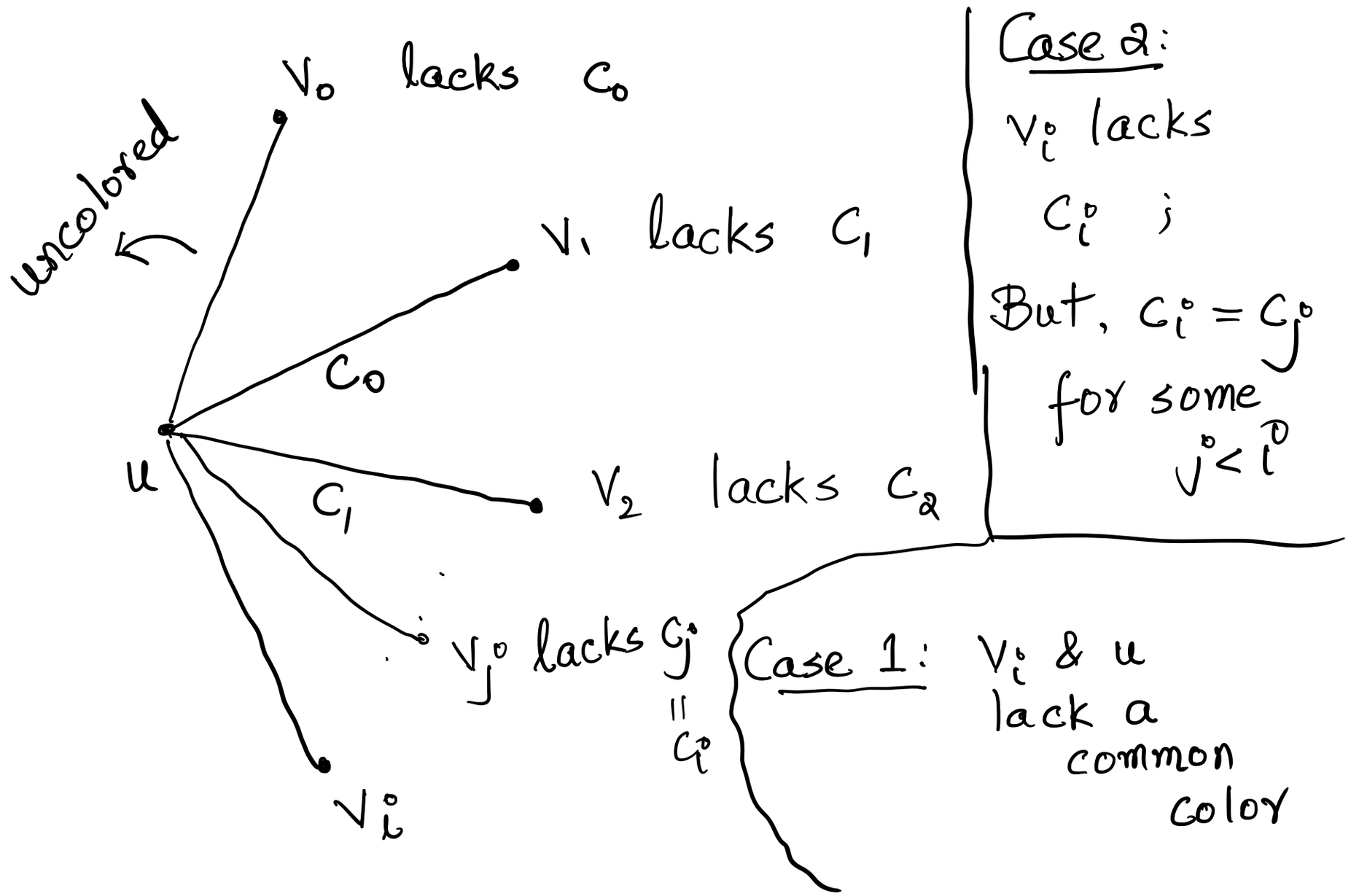
      $c_i \leftarrow c$ & break

   else

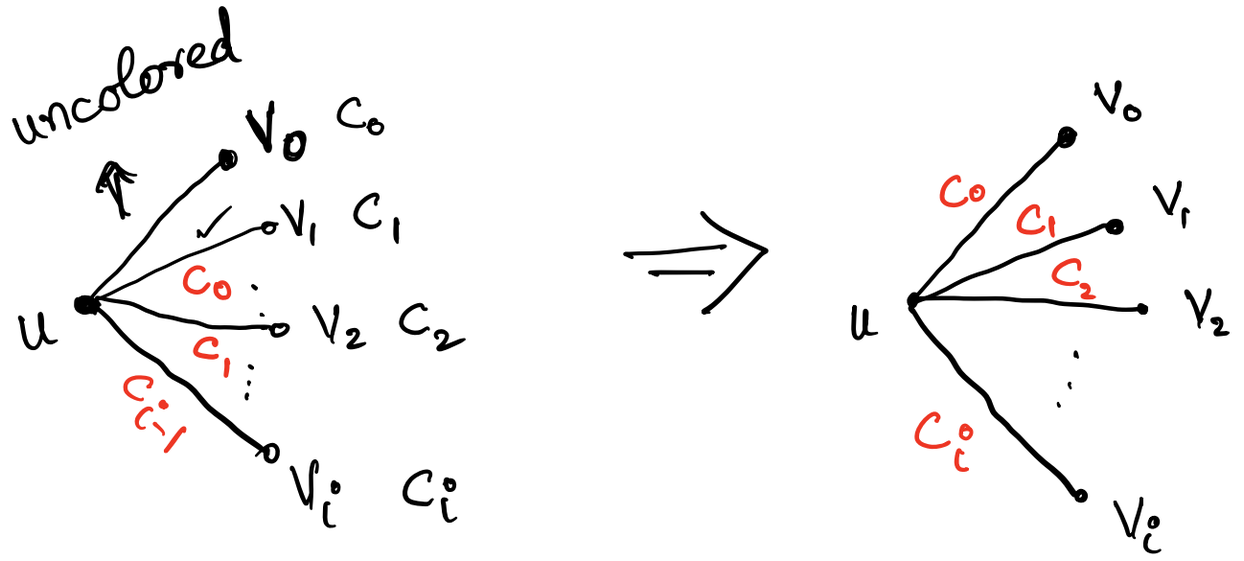
      $c_i \leftarrow$ some color that v_i lacks

      $v_{i+1} \leftarrow v'$, where (u, v') has color c_i

   until $c_i = c_j$ for some $j < i$

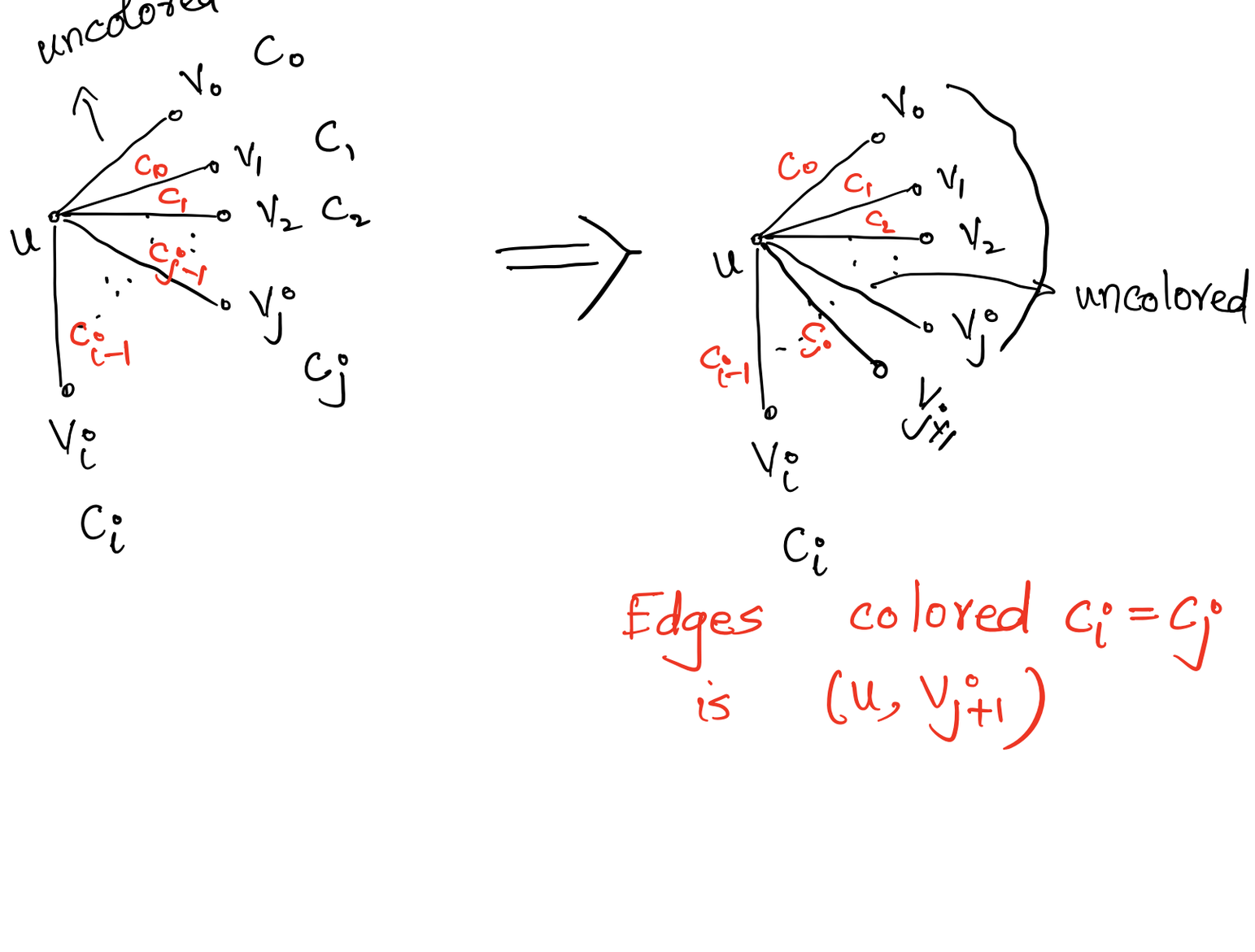

v_0 lacks c_0
uncolored
v_1 lacks c_1
c_0
u
c_1
v_2 lacks c_2
v_j lacks c_j
$= c_i$
v_i
Case 2:
v_i lacks c_i
c_i ;
But, $c_i = c_j$
for some $j < i$
Case 1: v_i & u lack a common color

3. if u & v_i lack color c_i then



4. else

(a) Let $j^\circ < i^\circ$ be such that $c_{j^\circ} = c_{i^\circ}$

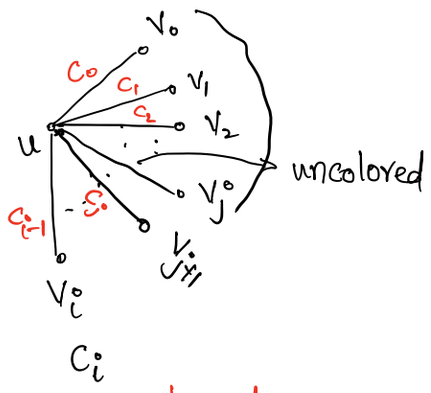


Edges colored $c_i = c_j$ is (u, v_{j+1})

(b) Pick color c_u that u lacks.

(c) Let E' be edges colored c_u & $c_i^\circ = c_j^\circ$ → endpoint of c_j°-colored edge

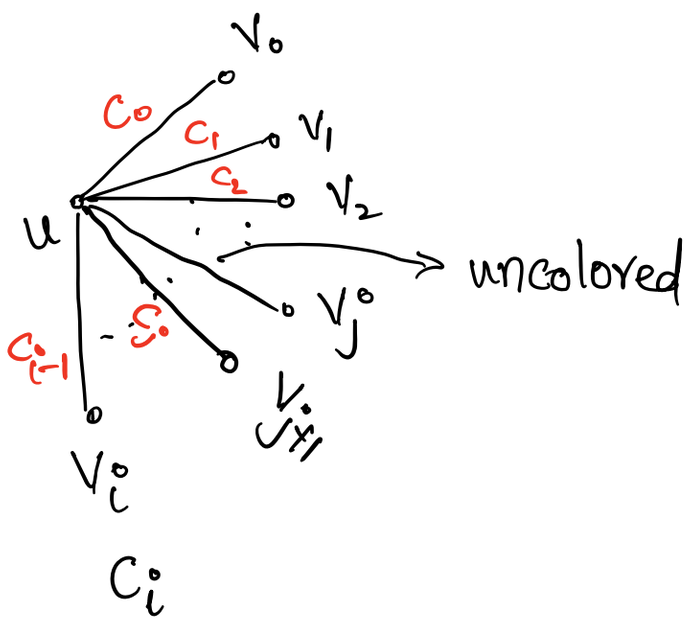
// u, v_i° & v_j° must be endpoints of some edges edge colored c_u in E'.



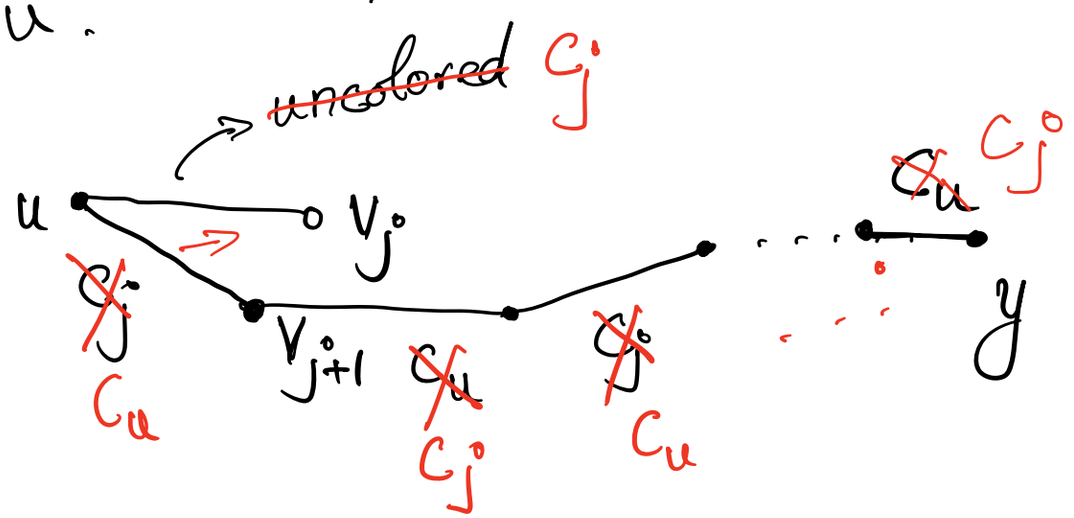
Edges colored $c_i = c_j^\circ$ is (u, v_{j+1}°)

* (V, E') is a collection of paths & cycles
* u, v_i°, v_j° are end points of paths

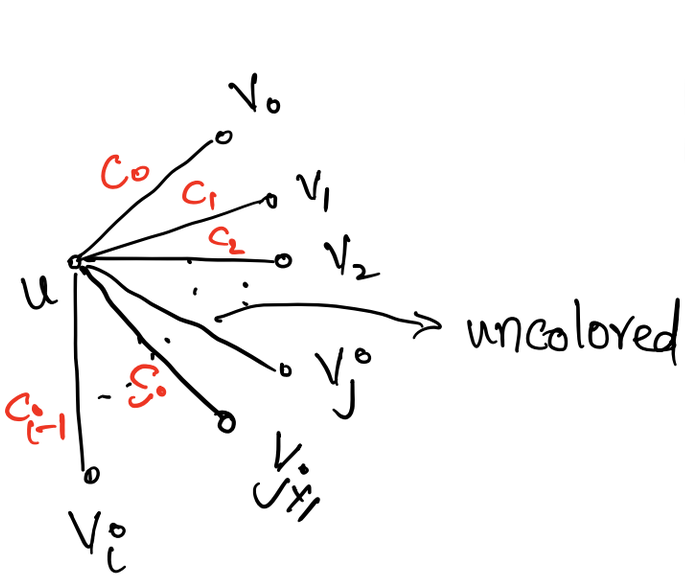
(d) if u & v_j in diff components of (V, E')

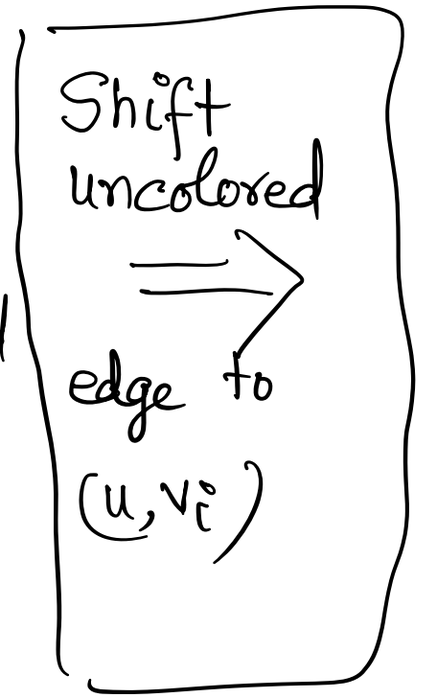


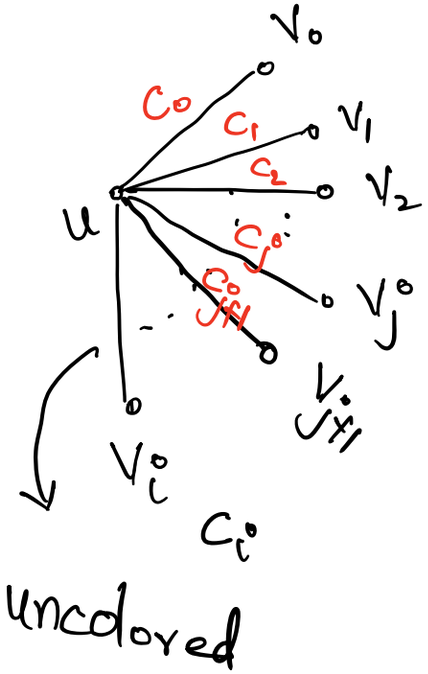
(a) Switch colors $c_i = c_j$ & c_u on the path containing u.

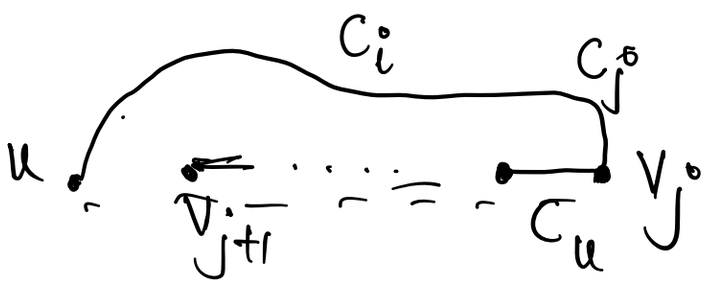


© if u & v_j in same component









* Switch colors c_i & c_u in component of u.

* Color (u, v_i) with c_i