

**Problem 1**

Prove that the 6 – Color – Rooted – Tree Algorithm produces a valid 6-coloring of a tree.

**Solution:** Let  $L_k$  denote the number of bits used to represent vertices at  $k$ -th iteration. Now for  $i = 1$  we have

$$L_1 = \lceil \log n \rceil + 1 \leq 2 \lceil \log n \rceil$$

Now let for  $i = k - 1$  we have  $L_{k-1} \leq 2 \lceil \log^{(k-1)} n \rceil$  and  $\lceil \log^{(k)} n \rceil \geq 2$ . Now

$$L_k = \lceil \log L_{k-1} \rceil + 1 \leq \left\lceil \log 2 \left\lceil \log^{(k-1)} n \right\rceil \right\rceil + 1 \leq 2 \left\lceil \log^{(k)} n \right\rceil$$

Hence if  $\lceil \log^{(k)} n \rceil \geq 2$  we have  $L_k \leq 2 \lceil \log^{(k)} n \rceil$ . Therefore the number of bits to represent the vertices decreases with each iteration and after  $O(\log^* n)$  many iteration  $L_k$  reaches the value of 3 (The limit  $L$  of  $\lim_{k \rightarrow \infty} L_k = \lim_{k \rightarrow \infty} \lceil \log L_{k-1} \rceil + 1$  is the solution of  $L = \lceil \log L \rceil + 1$ ). In those 3 bits the  $i_v$  takes 3 possible values and the  $b_v$  takes 2 possible values for each vertex  $v$ . Hence total number of colors is  $3 \times 2 = 6$ .

□

**Problem 2**

- Prove that every weakly connected component of a pseudoforest contains at most one cycle
- Find a 3 Coloring pseudoforest algorithm in  $O(\log^* n)$  time

**Solution:**

- Suppose there are two cycles  $C_1, C_2$  in a weakly connected component. Now  $C_1$  and  $C_2$  has to be disjoint cause other wise there will be a vertex in  $C_1 \cap C_2$  from which two edges have gone out side one for the next vertex in  $C_1$  and the other one for the next vertex in  $C_2$ . This is not possible since in a pseudoforest each vertex has out-degree exactly 1. So  $C_1$  and  $C_2$  are disjoint. Since they remain in same weakly connected component for  $u \in C_1$  and  $v \in C_2$  there exists a path  $u \rightsquigarrow v$  or  $v \rightsquigarrow u$ . WLOG let the path  $u \rightsquigarrow v$  exists. Let the path is  $P$ . Now there exists an edge  $(x, y) = e \in P$  such that  $e \notin C_1$  but the tail  $x$  of the edge is in  $C_1$ . Now since  $x \in C_1$  there is an edge going outward from  $x$  towards the next vertex in  $C_1$ . And also the edge  $(x, y)$  is going outwards along  $P$ . Hence out-degree of  $x$  is at least 2. Which is not possible. Hence every weakly connected component of a pseudoforest has at most one cycle.
- After applying 6 – Color – Rooted – Tree if there are more than 3 colors we shift down every color that is replace every non-root node's color with the color of the parent and color the root nodes with something different from the color before shifting and then for a color  $c$  we replace  $c$  with the smallest color other than  $c$ , the color of its child nodes and the color of the parent nodes (we can do this since there are more than 3 colors). In this shifting process total number of color is reduced by 1. Hence after 3 iteration of this shifting process number of colors is reduced to 3. In a shifting process the new coloring can be done for each vertex parallelly and removing the color  $c$  for each vertex can also be done in parallel. So each shifting process takes constant time. Since we do this shifting process 3 times it takes constant time to reduce the number of colors to 3. Hence to 3 color a tree it takes  $O(\log^* n)$  time.

Now for a pseudo forest we can take all vertices whose out-degree is 0. Then we apply the 3 – Color – Rooted – Tree algorithm where there is an edge from the vertex to its child vertices. So every vertices we check from where an edge is coming then color that according to the algorithm.

This algorithm will work because for any two vertices the parents of those vertices are disjoint cause otherwise there will be a parent vertex whose out-degree is more than 1. So this algorithm works. If there is an cycle then there is no sync vertex so we start with any vertex and apply the algorithm. Then at the end we check for the vertex to which edge from the chosen vertex goes and check the color of that vertex. Now the parents of the chosen vertex has same color so we can give the chosen vertex whose color is different from the colors of its parents and child.

□

### Problem 3

Is Maximum Independent Set for bounded degree graph  $NP - hard$ ?

**Solution:** It is not  $NP - hard$ . There is a polynomial time algorithm according to the paper “Independent Set on graphs with maximum degree 3” by Iyad A. Kanj, Fenghui Zhang

□