COL 106 Lecture 36

Topic : Testing Bipartiteness

Given graph G (in adjacency list representation), determine whether G is a bipartite graph.

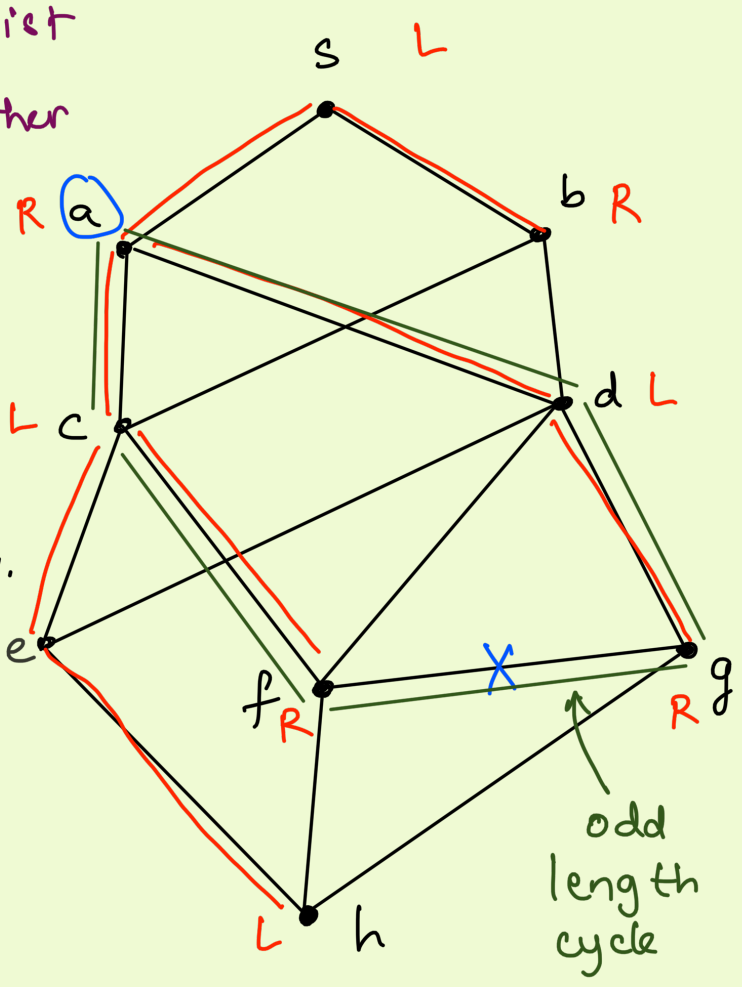
Perform BFT.

While processing the adj list of the dequeued vertex, say u:

If nbr v is not visited, enqueue v.

Else : if v and u are in the same level, return FALSE.

If BFT completes, return TRUE.



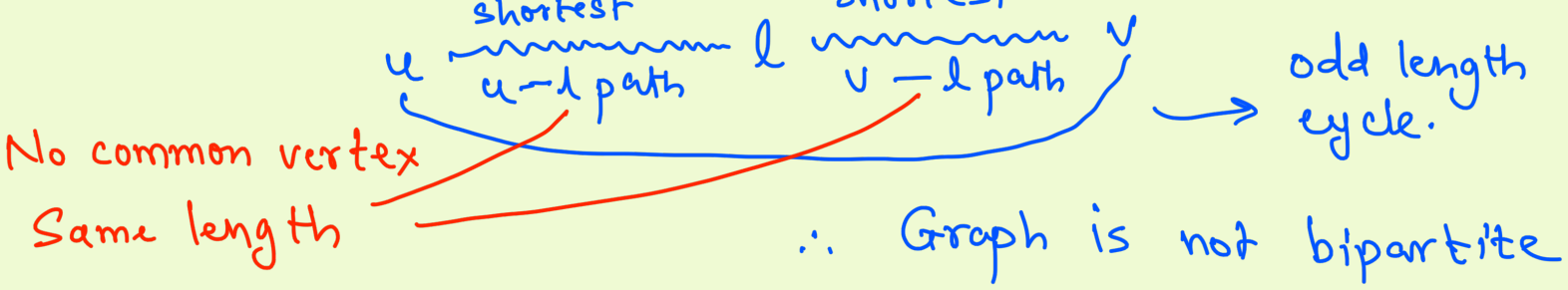
Proof of correctness of bipartiteness testing algorithm

Algorithm returns TRUE $\Rightarrow$ every edge between adjacent layers of BFT tree $\Rightarrow$ Graph is bipartite

(Odd levels on one side, even levels on the other)

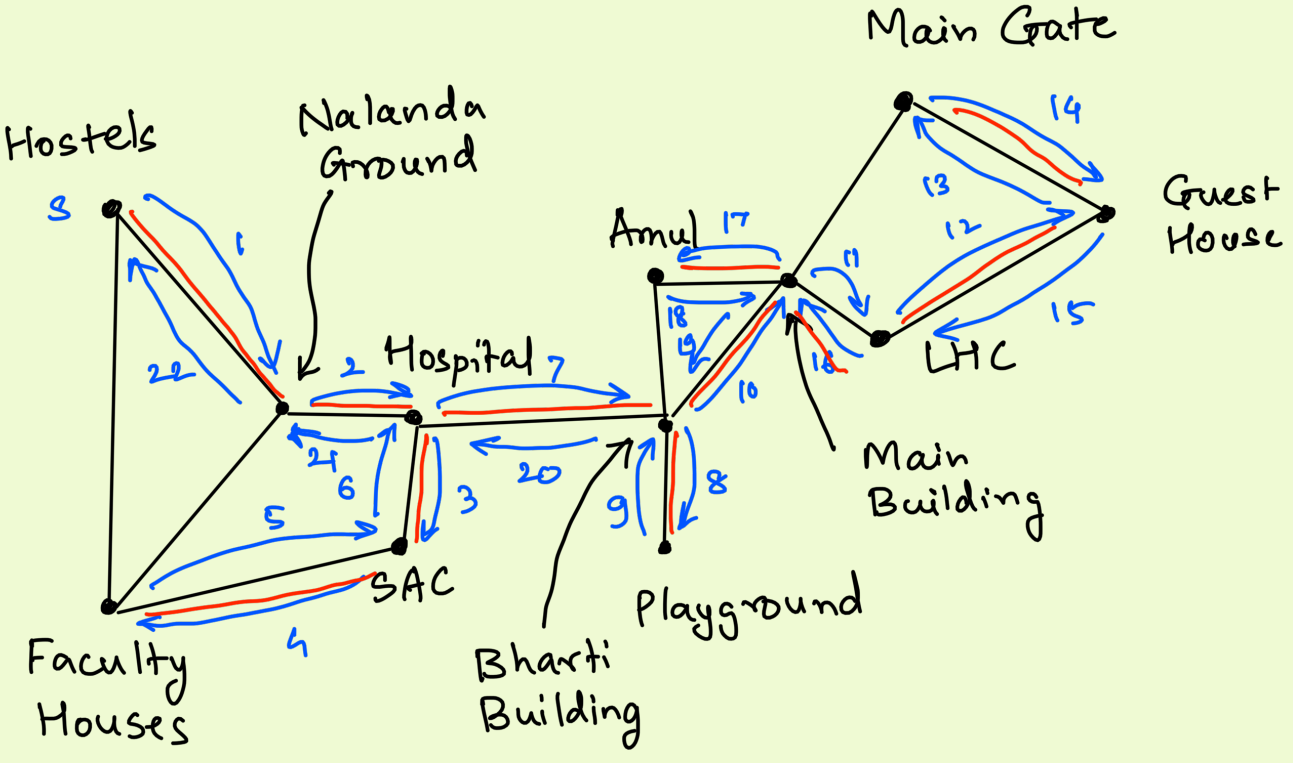
Algorithm returns FALSE $\Rightarrow$ $\exists$ edge, say $\{u, v\}$, where $dist(s,u) = dist(s,v)$.

Let l be "lowest common ancestor" of u, v on the BFT tree



$\therefore$ Graph is not bipartite

Depth-First Traversal.



visited : global array having one bool entry per vertex of G.

```
DFT (s):
    Set visited [V] ← FALSE for every vertex
    visited [s] ← TRUE
    DFT_rec (S)

DFT_rec (V):
    For each neighbor u of v
        If not visited [u]
            visited [u] ← TRUE
            prev [u] ← v
            DFT_rec (u)
```