Lab06

Xinhao Luo

May 12, 2020

1 Vertex cover problem (DPV 6.21)

Sub-problem Define arr[] represent whether the smallest vertex cover number include i. The arr[] include a tuple (x, y) where x represents the node i is not part of set and y represents the number node i is part of the set. We have the following equation:
$arr[i][0] = arr[i_1][1] + arr[i_2][1] + ... + arr[n][1]$
$arr[i][1] = arr[i_1][0] + ... + arr[i_n][0] + 1$

$1..n$ are children's node of i

Initialization

$arr[a][0] = arr[a][1] = 0$

Result The smallest number of element in the set should be $min(arr[n][0], arr[n][1])$ To get the set itself, do an DFS again over the graph. For each node has a smaller y, add it to the set

Order We need to go through each node's children first so we can avoid visit children multiple times. DFS will be the choice and since the graph is a tree, we can start by any node.

Explanation For each root of the subtree, either itself or all its children need to be in the smallest cover set so that all edges would be covered.

Runtime
1. The initialization of array takes $O(|V|)$
2. Running DFS is a linear time, $O(|V| + |E|)$ for the graph. (Using adjancent list implementation)
3. For each node, we may query its children, which means each node will be query twice. If we also need to get the set, we can run through the DFS again. However, runing DFS multiple time which does not affect the time complexity.
4. The overall runtime is linear, $O(|V| + |E|)$