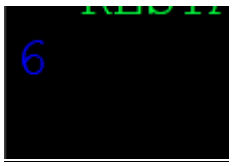
61. Minimum time to collect all apple in a tree

Code:

```
def minTimeToCollectApples(n, edges, hasApple):

    graph = [[] for _ in range(n)]
    for u, v in edges:
        graph[u].append(v)
        graph[v].append(u)

    visited = [False] * n

    def dfs(node):
        visited[node] = True
        total_time = 0

        for neighbor in graph[node]:
            if not visited[neighbor]:
                time_to_neighbor = dfs(neighbor)
                if time_to_neighbor > 0 or hasApple[neighbor]:
                    total_time += 2 + time_to_neighbor

        return total_time
    return dfs(0)

n = 7
edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]]
hasApple = [False, False, True, False, True, False, False]

print(minTimeToCollectApples(n, edges, hasApple))
```

Output:



Time Complexity:

- T(n)= O(n)