63. Remove Element

Code:

```
def maxIncome(edges, amount):
    import collections

    n = len(amount)
    if n == 1:
        return 0
    graph = collections.defaultdict(list)
    for a, b in edges:
        graph[a].append(b)
        graph[b].append(a)
    dp = [-float('inf')] * n

    def dfs(node, parent):
        if len(graph[node]) == 1 and graph[node][0] == parent:
            return amount[node]
        max_income = amount[node]
        for neighbor in graph[node]:
            if neighbor == parent:
                continue
            child_income = dfs(neighbor, node)
            if amount[node] >= 0:
                max_income += child_income / 2
            else:
                max_income -= amount[node] / 2
        dp[node] = max(dp[node], max_income)
        return max_income
    dfs(0, -1)
    return dp[0]
edges = [[0, 1], [0, 2], [2, 3], [2, 4], [4, 5]]
amount = [1, -2, 3, 4, -5, 6]
print(maxIncome(edges, amount))
```

Output:

```
1.875
```

Time Complexity:

- T(n)= O(n)