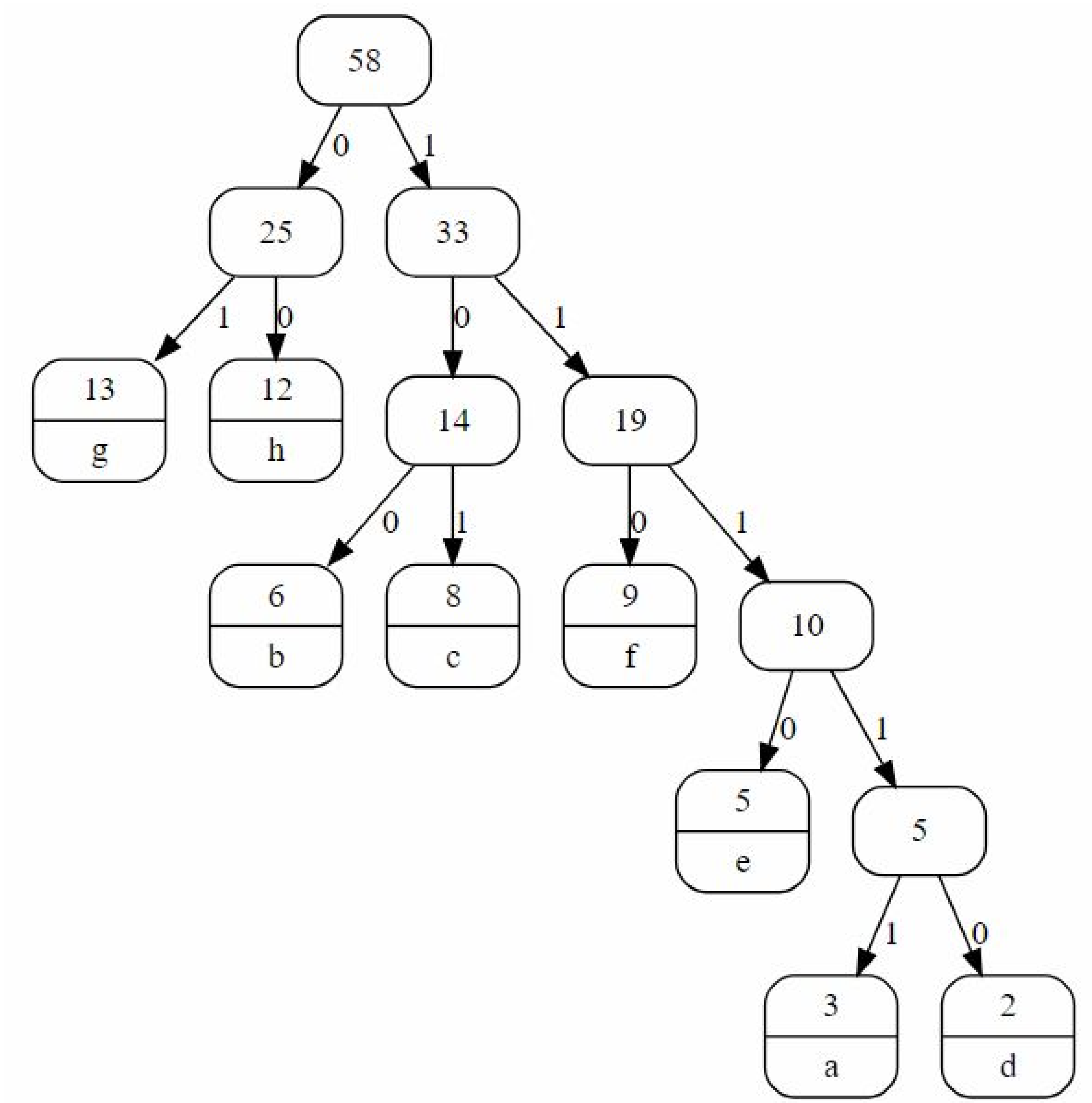
Solution 13

1.



2.

Greedy Solution:
Repeat until S = 0.
1. Find the largest coin denomination (say x), which is ≤ S.
2. Use ⌊ Sx⌋ coins of denomination x.
3. Let S = S mod x

Proof of Optimal Substructure: Let $\{c_1, c_2, \ldots, c_m\}$ be the coins comprising an optimal solution to the coin-changing problem, where c_i is either a penny, nickel, dime or quarter, and the c_i's sum to n, the change to be made. Putting aside the first coin choice c_1, the remaining coins $\{c_2, \ldots, c_m\}$ represent a solution to the n - $d(c_1)$ coin-changing problem, where $d(c)$ is the denomination of coin c.

If $\{c_2, \ldots, c_m\}$ were not an optimal solution to the n-$d(c_1)$ change problem, then there exists another solution $\{a_2, \ldots, a_k\}$, where $k < m$; thus, taking fewer coins. But if such a solution exists for n-$d(c_1)$, then combining it with c_1 would yield a better solution to the n change problem. Since we started with an optimal solution, this is a contradiction, $\{c_2, \ldots, c_m\}$ must be an optimal solution to the subproblem n-$d(c_1)$, and the coin-changing problem exhibits optimal substructure.

Proof of the Greedy Choice Property: Assume we have an optimal solution to the n-change problem $\{c_1, c_2, \ldots, c_m\}$. First, we know that the coins c_i add up to n no matter what order they are chosen, therefore, we can make any coin choice c_i swap with the first choice c_1 and still yield an optimal solution. We have only to prove that the greedy choice must reside somewhere in an optimal solution. Assume it does not. Then, one of the highest denomination coins less than n is not in the solution. But, that means that lower denomination coins must be used to obtain the number of cents of the higher-denomination coin. Since it takes at least two lower-denomination coins to equal the cents of a higher-denomination coin, we could always replace these coins with the greedy coin and obtain a better solution (fewer coins). Therefore, the greedy coin must be in the optimal solution. Thus, a greedy choice for the first coin will lead to an optimal solution.

Furthermore, since the greedy choice of coin c for the n-change problem reduces the problem to an n - $d(c)$ change problem, and the problem exhibits optimal substructure, the greedy algorithm will yield an optimal solution.

3.

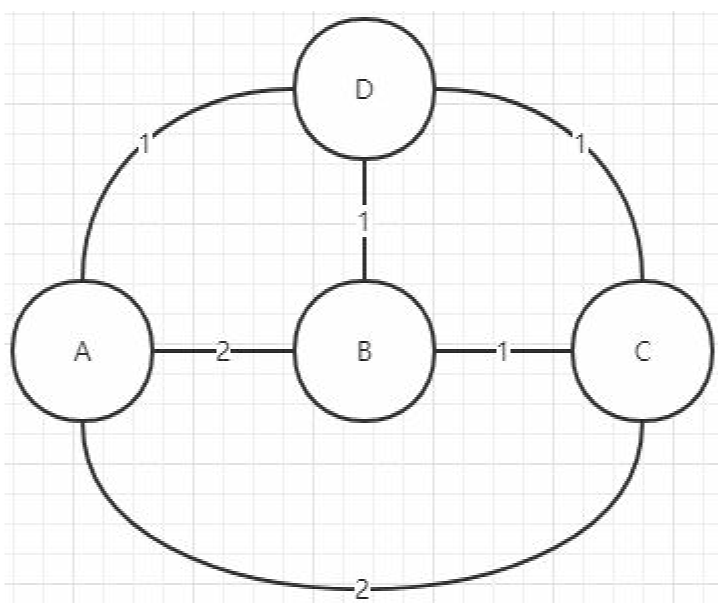
```
findMinimumDays(m)
        dist = n, count = 1
        for (i = n − 1 to 1)
                if (m[dist] − m[i] > 300)
                        if (m [i + 1] − m[i] > 300)
                                return -1
                        dist = i + 1
                        count ++
        return count
```

Time complexity is O(n), where n is the length of m array.

4.

No. A has only one path to out. If you start from A, you have to reach D next. But you have to go back to A to form a circle, so that you have to reach D again.

The graph that is augmented is as follows.

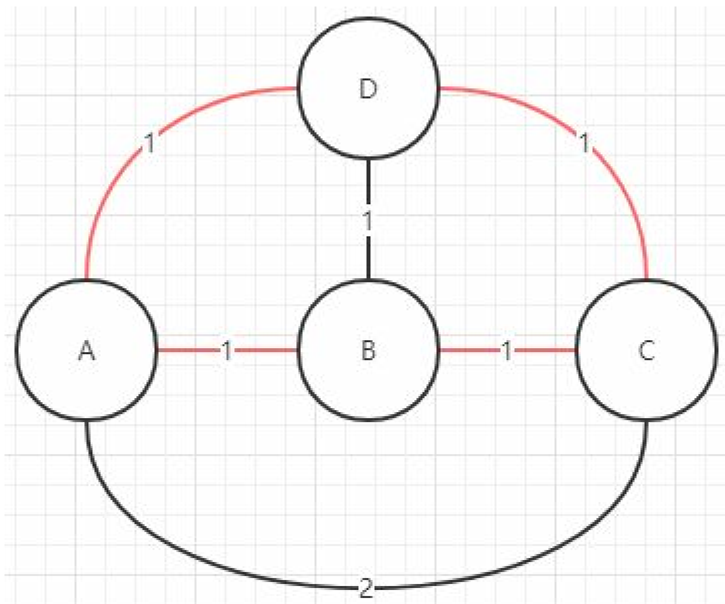


There is not solution for the TSP that has cost equal to 4.

5.

Yes.

The graph that is augmented is as follows.



There is a solution to the TSP that has a cost of 4.