1. Solution:

Let a is the number of quarters, b is the number of dimes, c is the number of nickels, d is the number of pennies. This problem is equivalent to given n, make $a + b + c + d$ is smallest such that $25a + 10b + 5c + d = n$. We can simply do this:

(a) Let $n\ mod\ 25 = n_1$, the remainder is n_1, which means $n = 25a_0 + n_1$, where $0 \le n_1 < 25$.

(b) Let $n_1\ mod\ 10 = n_2$, the remainder is n_2, which means $n_1 = 10b_0 + n_2$, where $0 \le n_2 < 10$

(c) Let $n_2\ mod\ 5 = n_3$, the remainder is n_3, which means $n_2 = 5c_0 + n_3$, where $0 \le n_2 < 5$

(d) Let $d_0 = n_3$.

Pseudocode:

```
GREEDY_CHANGE(n)
        n1 = n % 25
        a0 = (n – n1) / 25
        n2 = n1 % 10
        b0 = (n1 – n2) / 10
        n3 = n2 % 5
        c0 = (n2 – n3) / 5
        d0 = n3
        return a0, b0, c0, d0
```

The a_0, b_0, c_0, d_0 will be the value that satisfies the requirement of this problem, now we prove the correctness of this greedy algorithm:

(1) Optimal Substructure:

When we want to find a, b, c, d that satisfies $25a + 10b + 5c + d = n$, we try every possible combination of $n^{'} + n^{''} = n$. If we find the optimal solution for $n^{'}$ and $n^{''}$, we can find the optimal solution for n. In our greedy algorithm, every step we make our $n^{'} = tk_0$, where t is 25, 10, 5 or 1 and k_0 is a_0, b_0, c_0 or d_0 we discussed above. $n^{''} = n - n^{'}$, in every step, $n = n$, n_1, n_2 or n_3. As we can see, the optimal solution for sum up to n is the combination of the optimal solution for sum up to $n^{'}$ and $n^{''}$. So, this problem has optimal substructure property.

(2) Greedy Choice Property:

We prove this from button to up:

i. For any $n_2 = 5c + d$, the optimal solution must be to make $c = c_0$. If not, which means $c < c_0$, WLOG we can assume $c = c_0 - 1$, now, $d = d_0 + 5$. So,

$d + c = d_0 + 5 + c_0 - 1 = d_0 + c_0 + 4 > c_0 + d_0$. So, in the part (c) above, problem satisfies greedy choice property.

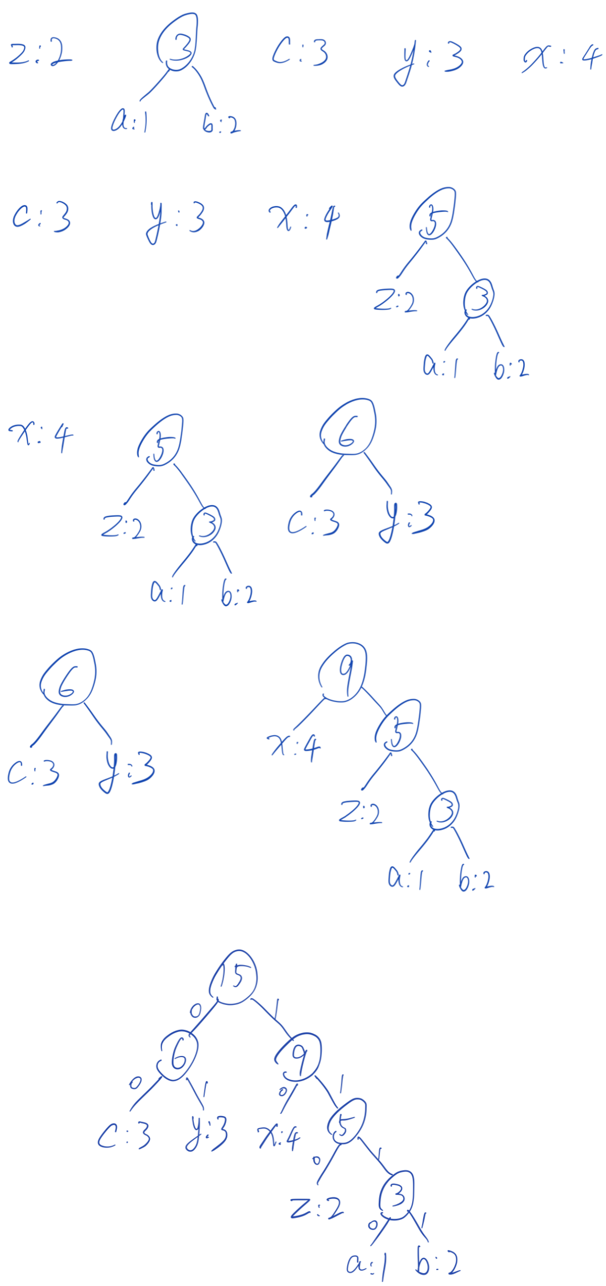
ii. For any $n_1 = 10b + n_2$, the optimal solution must be to make $b = b_0$. If not, which means $b < b_0$, WLOG we can assume $b = b_0 - 1$, now, $c = c_0 + 2$, $d = d_0$. So, $b + d + c = b_0 - 1 + c_0 + 2 + d_0 = b_0 + c_0 + d_0 + 1 > b_0 + c_0 + d_0$. So, in the part (b) above, problem satisfies greedy choice property.

iii. For any $n = 25a + n_1$, the optimal solution must be to make $a = a_0$. If not, which means $a < a_0$, WLOG we can assume $a = a_0 - 1$, now, $b = b_0 + 2$ $c = c_0 + 1$, $d = d_0$. So, $a + b + d + c = a_0 - 1 + b_0 + 2 + c_0 + 1 + d_0 = a_0 + b_0 + c_0 + d_0 + 2 > a_0 + b_0 + c_0 + d_0$. So, in the part (a) above, problem satisfies greedy choice property.

So, we have proved greedy choice property and optimal substructure for this problem, every step is a safe greedy choice, so our algorithm will work. The running time will $O(1)$ because we only need 4 steps to find a, b, c and d.

2. 38.

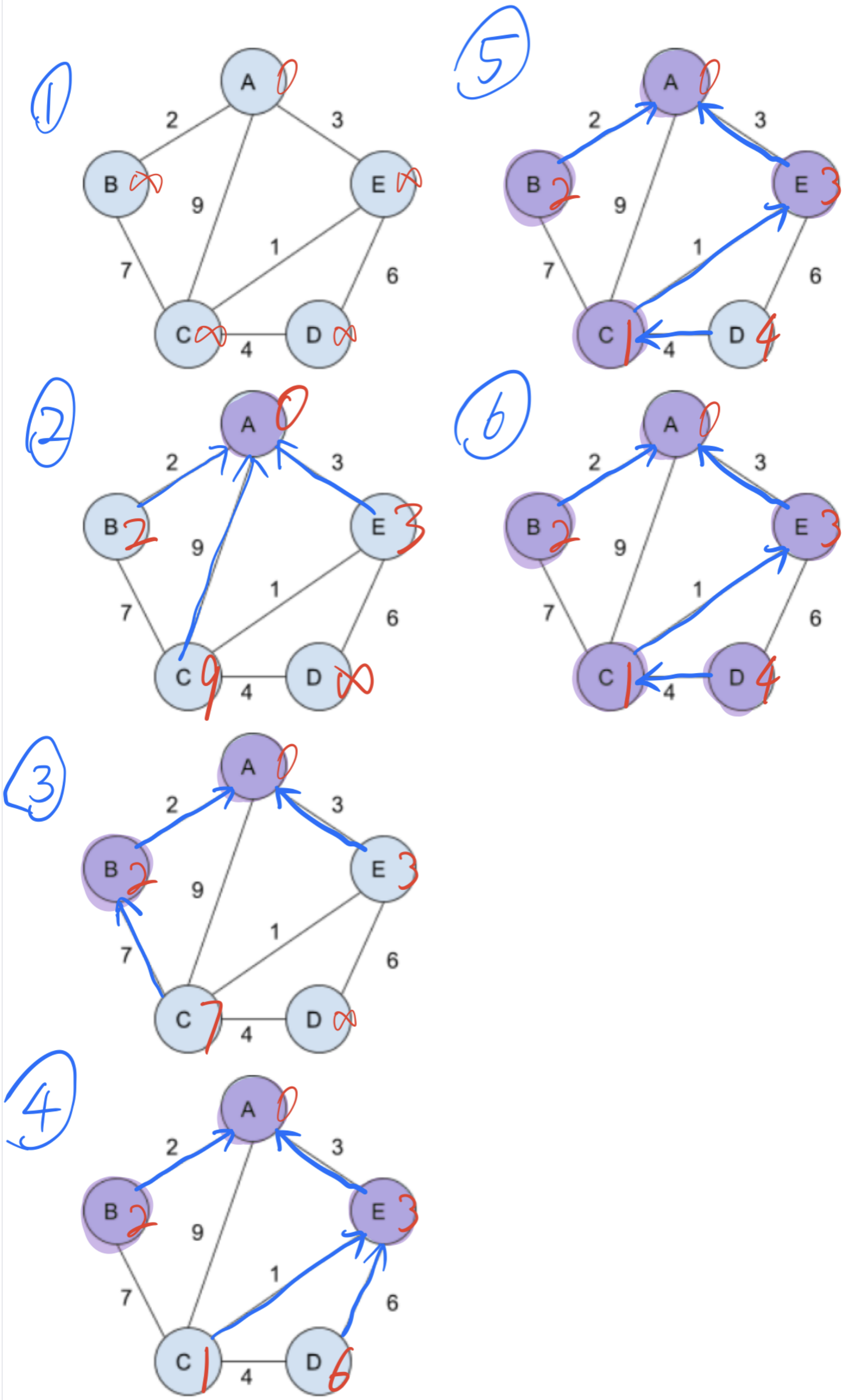
a:1 b:2 z:2 c:3 y:3 x:4



$2\times3+2\times3+2\times4+3\times2+4\times1+4\times2$

$= 38$

3.



1
A 0
B ∞
C ∞
D ∞
E ∞
2 3 9 1 7 6 4
2
A 0
B 2
C 9
D ∞
E 3
2 3 9 1 7 6 4
3
A 0
B 2
C 7
D ∞
E 3
2 3 9 1 7 6 4
4
A 0
B 2
C 1
D 6
E 3
2 3 9 1 7 6 4
5
A 0
B 2
C 1
D 4
E 3
2 3 9 1 7 6 4
6
A 0
B 2
C 1
D 4
E 3
2 3 9 1 7 6 4

4.

1. {C, D}
2. {A, E}
3. {A, C}
4. {A, B}