Answer to the sample test questions

Question 1

a) i. T ii. F iii. T iv. T

b) i. $T(n) = \Theta(n^3)$ ii. $T(n) = \Theta(n^2)$

c) i. unconditionally true ii. C is NP-Complete

Question 2

(a) combine or the same, depends on part (b).

(b) & (c) len_of_left_side:

- If it is implemented below, answer to part (a) is “combine”
- If it is implemented as shown on the right, answer to part (a) is “the same”

O(1)

```
def len_of_left_side(n):
    if n == 0:
        return 0, 0
    b = math.floor(math.log2(n+1))
    size_left = min(2 ** b - 1,
                    n - 2 ** (b-1) + 1)
    return size_left
```

O(log n)

```
def incr_size(lvl, size, remain):
    if remain > 2 ** lvl:
        return size + 2 ** lvl, remain - 2 ** lvl
    else:
        return size + remain, 0

def len_of_left_side(n):
    size_left, size_right = 0, 0
    i, rem = 0, n
    while rem > 0:
        size_left, rem = incr_size(i, size_left, rem)
        size_right, rem = incr_size(i, size_right, rem)
        i += 1
    return size_left
```

Question 2(d)

- No matter whether we use the O(1) or the O(log n) algorithm for divide step, we need O(log n) for the combine step.
- Therefore, $T(n) = T(a \cdot n) + T(b \cdot n) + c \cdot \log n$ where $a + b = 1$
  - where a·n, b·n are the sizes of the left sub-tree and right sub-tree
- If the two subtrees are balanced, we have $T(n) = 2T(\frac{n}{2}) + c \cdot \log n$ by Master's theorem, $T(n) = \Theta(n)$
- If the two subtrees are not balanced, it is at the worst case

$$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + c \cdot \log n \leq 2T(\frac{2n}{3}) + c \cdot \log n$$

- By master's theorem, $T(n) = O(n^{\log_{\frac{3}{2}} 2}) = O(n^{1.71})$

Question 2(d)

- In fact, we can show $T(n) = \Theta(n)$
- Assume $T(n) = d \cdot n - c \cdot \log n$ for some constant d
- Then $T(a \cdot n) = d \cdot a \cdot n - c \cdot \log(a \cdot n)$ and $T(b \cdot n) = d \cdot b \cdot n - c \cdot \log(b \cdot n)$

$$\begin{aligned} T(n) &= T(a \cdot n) + T(b \cdot n) + c \cdot \log n \\ &= d \cdot a \cdot n - c \cdot \log(a \cdot n) + d \cdot b \cdot n - c \cdot \log(b \cdot n) + c \log n \\ &= d \cdot n - c \cdot \log n - c \cdot \log ab \\ &\leq d \cdot n - c \cdot \log n \end{aligned}$$

- Therefore, we have $T(n) = d \cdot n - c \cdot \log n = \Theta(n)$

Question 3

a) Subproblem: number of ways to make up a change of k cents using dominations $\{d_1, d_2, \ldots, d_j\}$ where j < m

b) Yes

c) $$f(n,m) = \sum_{0 \le k \le n \text{ and } d_m | n-k} f(k, m-1)$$

d) Let p=LCM(d_{m-1}, d_m), both f(n,m-1), f(n-p,m-1) uses f(n-p, m-2)

e) $O(n^2 m)$

f) The programme itself gives a counter example, with denominations [1, 2, 5] and target 10, the number of ways should be 10.

Question 4

a) Yes

b) The two parts with “ if   >= current_best” is redundant

c) Worse case: $O(2^n)$

d) Worse case scenario: L = sum(w), need to consider all combinations

e) Refer to class notes on 0/1 Knapsack problem

f) 390

Question 5

- This is Josephus problem: en.wikipedia.org/wiki/Josephus_problem

- (a): Implement a doubly linked circular list, each node is linked to left and right. Counting out is implemented by node deletion. Implementation omitted.

- (b): Let f(n-1) be the position of the survivor in the circle of n-1 astronauts
- The suicided astronaut is at the immediate front of the astronaut who counted 1 in this round of n-1 astronauts
- If we now look at the circle of n astronauts by adding the suicided one back, the one counted x (in the circle of n-1) will count x+m in the circle of n, therefore, the survivor will count f(n-1)+m, thus f(n) = [f(n-1)+m]%n
- f(1) = 0, when m = 5, f(2) = 1, f(3) = 0, f(4) = 1, f(5) = 1, f(6) = 0, f(7) = 5, f(8) = 2