CSC148 - Interview Questions

In this exercise, we will solve a few problems that are some popular interview questions asked by tech companies. You'll see how the concepts and skills that you learned in CSC148 can help with answering these questions.

Question 1: Linked List Cycle

Given `head`, the head of a linked list, determine if the linked list has a cycle in it.
(Source: `https://leetcode.com/problems/linked-list-cycle/`)

```
class ListNode(object):
    def __init__(self, x, next=None):
        self.val = x
        self.next = next

def has_cycle(head: ListNode) -> bool:
    """
    returns if the linked list starting from <head> has a cycle.

    >>> lst = ListNode(1, ListNode(4, ListNode(8)))
    >>> has_cycle(lst)
    False
    >>> lst.next.next.next = lst
    >>> lst2 = ListNode(3, lst)
    >>> has_cycle(lst2)
    True
    """
    pass
```

Question 2: Daily Temperature

Given a list of daily temperatures `temps`, return a list such that, for each day in the input, tells you how many days you would have to wait until a warmer temperature. If there is no future day for which this is possible, put 0 instead.
(Source: `https://leetcode.com/problems/daily-temperatures/`)

```
def daily_temperatures(temps: List[int]) -> List[int]:
    """
    Given a list of daily temperatures <temps>, return a list such that, for
    each day in the input, tells you how many days you would have to wait until
    a warmer temperature. If there is no future day for which this is possible,
    put 0 instead.

    >>> lst = [73, 74, 75, 71, 69, 72, 76, 73]
    >>> daily_temperatures(lst)
    [1, 1, 4, 2, 1, 1, 0, 0]
    """
    pass
```

Question 3: Square Root

Given a non-negative integer x, compute and return the square root of x. Since the return type is an integer, the decimal digits are truncated, and only the integer part of the result is returned. **Requirement:** You are NOT allowed to use the built-in `sqrt()` method or the power operator (**), and the runtime of your algorithm must be in $\mathcal{O}(\log x)$.
(Source: `https://leetcode.com/problems/sqrtx/`)

```
def my_sqrt(x: int) -> int:
    """
    Compute and return int(sqrt(x))
    Precondition: x >= 0

    Requirements:
    - You must NOT use any given sqrt() method or the power operator (**)
    - The runtime must be O(log x)

    >>> my_sqrt(0)
    0
    >>> my_sqrt(16)
    4
    >>> my_sqrt(48)
    6
    """
    # Below is an inefficient solution with runtime O(sqrt(x))
    # You need to improve it to O(log x)
    for ans in range(x+1):
        if ans * ans <= x and (ans + 1) * (ans + 1) > x:
            return ans
    return 0
```