

Q.1 You are given a sorted list of 15 unique integers. Create a list such that the worst-case scenario for binary search would take 4 steps to locate an element. Explain your reasoning.

Q.2 Determine the time-complexity of function F1 which performs some operation on a list. The time-complexity of function F2 is reported to be $O(n)$.

```
def F1(lst):
    if len(lst) <= 1:
        return 0
    else:
        F2(lst)
        mid = len(lst)//2
        return F1(lst[:mid]) + F1(lst[mid:])
```

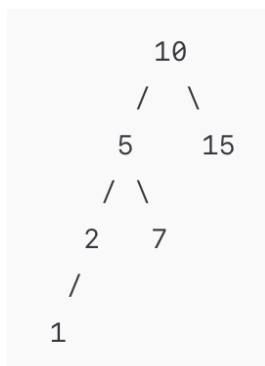
Q.3 You are given a dictionary d where each key is an integer, and each value is:

1. A list of dictionaries.
 - Each dictionary has:
 - Keys as strings.
 - Values that can be:
 1. A tuple of two integers.
 2. A list where one element is a string and the rest are integers.

Write a list comprehension to extract all unique integers from the structure, but only include integers that are greater than 10.

Q.4 Write a generator function `leftmost_and_parent_cycle(bst_root)` that:

1. Traverses down the leftmost path of a binary search tree (BST) starting at the root.
2. After reaching the leftmost node, yields values of the parent nodes in reverse order (going back up to the root).
3. Repeat the process infinitely.



```
it = leftmost_and_parent_cycle(bst_root) # Iterating through the generator
print(next(it)) # Output: 10 (downward)
print(next(it)) # Output: 5 (downward)
print(next(it)) # Output: 2 (downward)
print(next(it)) # Output: 1 (leftmost node)
print(next(it)) # Output: 1 (reverse upward)
print(next(it)) # Output: 2 (reverse upward)
print(next(it)) # Output: 5 (reverse upward)
print(next(it)) # Output: 10 (reverse upward)
print(next(it)) # Output: 10 (start again)
```

Q.5 Write a higher-order function `powerChecker(power)` that:

1. Takes an integer power (greater than or equal to 1) as input.
2. Returns a function that takes an integer num as input and returns:
 - True: if num is a perfect power-th power (i.e., there exists an integer k such that $k^{\text{power}} = \text{num}$).
 - False: otherwise.