Problem1:
Used Linked List data structure. Insertion and removal is constant O(1)
We are using Dictionary, So space Complexity will be O(n)

Problem 2

Time Complexity(O^x) X is exponential as we don’t know how many subfolders will be there. For that n number of subfolder in the directory n times, the for loop will be called.
Space: O(n log n) Depends on how many directories to recursively.

Problem 3

Have used python heapq library.
Time Complexity for this data structure will be O(n)
Space: O(n log n)

Problem 4
Same Recursvie Solution as in problem 2

Time Complexity(O^x)
Space: O(n log n)

Problem 5
Basically, a Linked List implemented.
Time Complexity O(1)
Space Complexity O(n)

Problem 6

Union
Time Complexity O(n) One iteration of a set
Space Complexity O(n) as Linked List implementation

Intersection
Time Complexity O(n^2) Two iteration of a set
Space Complexity O(n) as Linked List implementation