Assignment 9 - Complexity

1. Take as input N, a number. Take N more inputs to form an integer array. Find the max and min of the array in O(n) time complexity and constant space complexity.
2. Take as input N, a number. Take N more inputs to form an integer array. Write a function which tests if the array has any duplicates or not and returns a Boolean value. Print the value returned. Target time complexity is O(nlogn)
3. Take as input N, a number. Take N more inputs to form an array. Take as input M, a number. Take M more inputs to form an array. Write a function which returns intersection of both arrays. Target complexity is O((n + m)logm)) where m is the size of smaller array.
4. Take as input N, a number. Take N more inputs to form an array. Take as input M, a number. Take M more inputs to form an array. Take as input P, a number. Take P more inputs to form another array. Write a function which returns intersection of three arrays. Target complexity is O((n + m + p)logp)) where p is the size of smallest array.
5. Take as input x and n, two numbers. Write a function to calculate x raise to power n. Target complexity is O(logn).
6. Take as input N, a number. Write a recursive function to calculate Nth Fibonacci. Target complexity is O(N).
7. Take as input N, number of stairs. A person could climb 1, 2 or 3 stairs at a time. Write a recursive function to calculate the different ways these stairs could be climbed. Target complexity is O(N).
8. Take as input N, size of a snakes and ladders board. A player throws a dice and moves as many places as is the value on dice. Write a recursive function to calculate the different ways to cross the board. Target complexity is O(N).
9. Take as input N1 and N2, both numbers. N1 and N2 is the number of rows and columns on a rectangular board. Our player starts in top-left corner of the board and must reach bottom-right corner. In one move the player can move 1 step horizontally (right) or 1 step vertically (down). Write a recursive function which returns the count of different ways the player can travel across the board. Print the value returned. Target complexity is O(N1 * N2).
10. Take as input x and n, two numbers. Calculate the value of following polynomial.

    $$\mathbf{1.X^{n} + 2.X^{n-1} + 3.X^{n-2} + .. + n.X^{1}}$$

    Target complexity is O(N)

11. Take as input str, a string. Write a function which returns the number of palindromic substrings in the string. Target complexity is $O(n^2)$.
12. Take as input N, a number. Print prime numbers until N. Target complexity is O(nlog(logn)).

Assignment 9 - Complexity

13. Take as input str, a string. Find the character with highest frequency in the string and print it. Target complexity is O(n).
14. Take as input N, a number. Take N more inputs to form an array. Take as input target, a number. Print all pairs in N which sum to target. Target complexity is O(nlogn).
15. Take as input N, a number. Take N more inputs to form an array. Take as input target, a number. Print all triplets in N which sum to target. Target complexity is O(n^2).
16. Take as input N, a number. Take N more inputs to form an array. Reorder items of array in such a manner that all 0's are moved to end. Target complexity is O(n)
17. Take as input N, a number. Take N more inputs to form an array. The array contains only 0 and 1. Sort the array in a single scan.
18. Take as input N, a number. Take N more inputs to form an array. The array contains only 0, 1 and 2. Sort the array in a single scan.
19. Take as input N, a number. Take N more inputs to form an array. The array was supposed to contains values from 0 to N -1 (no duplicates). Accidentally, a number got duplicated. Find the duplicate number. Target complexity is O(N).
20. Take as input N, a number. Take N more inputs to form an array. The array was expected to be sorted. Accidentally, it got k items forward rotated. E.g. 3, 4, 5, 6, 1, 2 has been forward rotated 4 times. Find K. Target complexity is O(N).