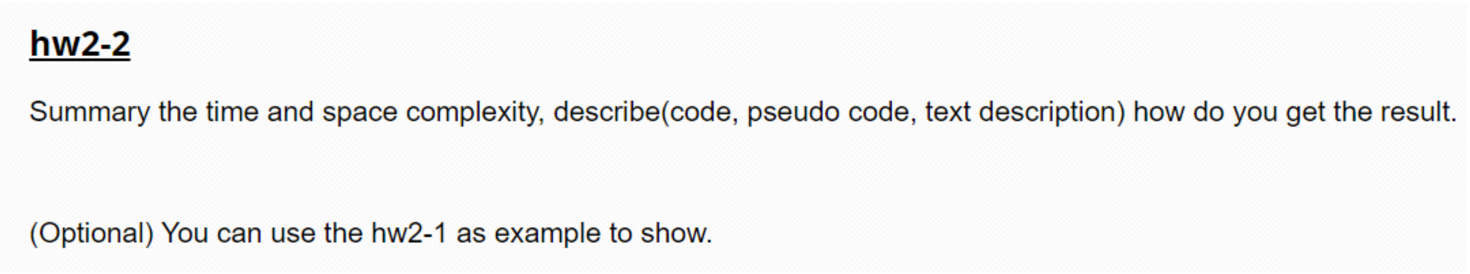
# Hw 2-2

Monday, November 6, 2023 4:32 PM



**hw2-2**

Summary the time and space complexity, describe(code, pseudo code, text description) how do you get the result.

(Optional) You can use the hw2-1 as example to show.

Time Complexity - the amount of time it takes for the algorithm to run as a function of the length of the input

Space Complexity - the amount of space it takes up for the algorithm to run as a function of the length of the input

Both Time and Space Complexity is expressed by Big-O notation.

The time complexity to execute one operation is constant time. Total operations for an input length on N are calculated. Steps to calculate time complexity:

1) Determine what the input and N represents
2) Express the maximum number of operations that the algorithm performs, in terms of N
3) Eliminate all excluding the highest order terms
4) Remove all the constant factors

Time Complexities (from fastest to slowest):

- Constant: O(1)
- Logarithmic: O(log(N))
- Linear: O(N)
- Superlinear: O(N log(N))
- Quadratic: O(N^2)
- Exponential: O(2^N)
- Factorial: O(N!)

Space complexity represents the amount of memory needed to completely run the algorithm. To determine the space complexity, one must focus on two vital parts:

1) Fixed part - independent of the input size; includes memory for instructions, constants, variables, etc.
2) Variable part - dependent on the input size; memory for recursion stack, referenced variables, etc.