Assignment 1

Mingyuan Ba
SID: 530157791

March 9, 2024

Problem 1

a)

The number of iterations is

$$(n-1)+(n-2)+\cdots+1+0=\frac{n(n-1)}{2}$$

which is bounded by n^2, For each iteration corresponding to indices i, j, we only need to perform a single comparison operation, so the time complexity per iteration is $O(1)$. Thus , the overall time complexity is $O(n^2)$.

b)

Assume for simplicity that n is even. To lowerbound the running time, consider only comparisons made during the first half of its execution. Since this is part of the full execution, analyzing only this part gives a lower bound on the total running time. The main observation we need is that for each of the considered iterations, we make at least $\frac{n}{2}$ comparisons, allow us to lower bound the total number of comparisons made:

$$\sum_{i=0}^{n-1}(n-1-i) \geq \sum_{i=0}^{\frac{n}{2}-1}\frac{n}{2}=\frac{n^2}{4}=\Omega(n^2)$$

Problem 2

Sum()

When a new element get pushed or poped, the sum need to be updated. The **Sum** operation return the value of variable **sum**, which takes O(1) time.
We modified the push and pop operations. Adding the new element to the sum takes O(1) time, so push still runs in O(1) time. Similarly, subtracting the removed element from the sum takes O(1) time, so pop still runs in O(1) time.

```
1: function NewPush(e)
2:     sum ← sum + e
3:     push(e)
4: end function
```

```
1: function NewPop
2:     e ← pop
3:     sum ← sum − e
4: end function
```

```
1: function Sum
2:     return sum
3: end function
```

Problem 3

a)

```
1: function A(B, m)
2:     n ← len(B)
3:     j ← n − 1
4:     i ← 0
5:     counts ← 0
6:     while i < j do
7:         if B[i]] + B[i] ≥ m then
8:             counts ← counts + (j − i)
9:             j ← j − 1
10:        else
11:            i ← i + 1
12:        end if
13:    end while
14:    return count
15: end function
```

b)