CS521 Homework 4
Name: Yuxiao Wu
2021/11/6
No collaborators, No late days
Source: textbook

Problem 1 (15 points)

Give the Big O run-time of the following algorithms. Explain/show your work.

Binary Search:

```
def binary_search(arr, low, high, x):
    # Check base case
    if low > high:
        return None
    else:
        mid = (high + low) // 2
        element = arr[mid]

        if element == x:
            return mid
        elif element > x:
            return binary_search(arr, low, mid - 1, x)
        else:
            return binary_search(arr, mid + 1, high, x)
```

Each iteration will half the length of arr, which is log(len(arr)) iterations
Run time = O(Log(N))

Selection Sort:

```
def selection_sort(arr):
    for i in range(len(arr)):
        smallest_index = i
        smallest_value = arr[i]

        # Find smallest element
        for j in range(i, len(arr)):
            candidate = arr[j]
            if candidate < smallest_value:
                smallest_index = j
                smallest_value = candidate

        # Swap front with smallest value
        temp = arr[i]
        arr[i] = smallest_value
        arr[smallest_index] = temp

    return arr
```

N

N

Nested for loop,
Runtime = $O(N^2)$

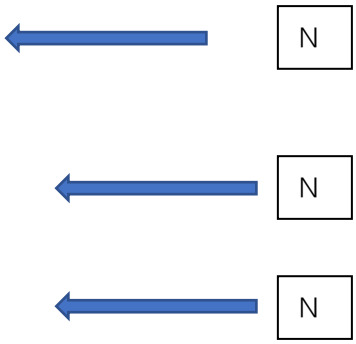
Insertion Sort:

```
def insertion_sort(arr):

    for i in range(len(arr)):
        current_element = arr[i]
        # Find correct place in list
        j = 0
        while j < i and arr[j] <= current_element:
            j += 1

        # Move all larger elements over 1
        for x in range(i, j-1, -1):
            arr[x] = arr[x-1]

        # Insert current element
        arr[j] = current_element
```



While and for loop series inside outer for loop
Runtime = O(N*2N) = $O(N^2)$

Problem 2 (10 points)

Simplify the following Big O expressions. Show work for partial credit.

1. $O(2 * N + Log(N) + N^3 + 10)$
2. $O(1000)$
3. $O(10 + 20 * N + 10 * N * Log(N) + 10 * N^2)$
4. $O(N * Log(N) + 20 * N + N * Log(N)^2)$
5. $O(2 * N^2 * Log(N))$

1. = $O(N^3)$
2. = O(C)
3. = $O(N^2)$
4. = $O(N*Log(N)^2)$
5. = $O(N^2*Log(N))$

Problem 3 (15 points)

What are the 4 tenants of Object-Oriented Programming with brief explanations (<2 sentences) of each?

1. Encapsulation: Encapsulation is the mechanism of hiding of data implementation by restricting access to public methods. Instance variables are kept private and accessor methods are made public to achieve this.
2. Abstraction: Abstract means a concept or an idea which is not associated with any instance. Using abstract class/Interface we express the intent of the class rather than the actual implementation.
3. Inheritance: Using Inheritance, In derived classes we can reuse the code of existing super classes.
4. Polymorphism: By using polymorphism, We can write a code that works on the superclass, and it will work with any subclass type as well