Problem4:
The overall time complexity of my algorithm is O(n) + the time it takes to bubble sort which would be, O(n) + O(n^2) which would just be equal to O(n^2).

O(n):

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

O(n^2):

```
void Problem4A(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)


        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}
```

There cannot be a linear time solution for this problem because you need to pass information from one function to another and in that function you need to run the previous function in order to sort, for example in my code I need to run the swap function in order to run the main function. It can't be linear because there is no such algorithm that sorts arrays in linear time.