Algorithms and Data Structures
Homework5

bessefiany

March 15, 2021

1 Problem 5.1

1.1 a

```
Bubble Sort(A,size)
        /*checks even for worst case*/
        for i=0 to size-1:
                for j=0 to size-i-1:
                        /* if this pair is out of order */
                        if(A[j]> A[j+1])
                                swap(A[i - 1], A[i])
                                        temp=A[j]
                                        A[j]=A[j+1]
                                        A[i+1]=temp
```

1.2 b

Worst case::
The worst case scenario is obviously the descending ordered array.
Therefore, the comparison and swapping will be done the maximum time $(n-1)$, then second iteration, it will be done $(n-2), (n-3), (n-4), ...2, 1$.
So, we have a running time of:
$(n-1) + (n-2) + (n-3) + ... + 2 + 1 = (n(n+1))/2$
That is to say $O(n^2)$
Average case::
AN average case scenario is when the array is arbitrary sorted,
though we should go through both loops, with $(n-1)$ comparison each,
and so on... we'll assume that we have a probability of only half swaps,
Therefore we have:
$(n/2)(n-1)/2 = (n(n-1))/2 * 2$
which still $\Theta(n^2)$
Best case::

In the best case we have an already sorted array, so we don't need to go through the inner
loop, So, I'll assume that the program will run n-1 times,so , Id say: $\Omega(n)$

1.3 c

see attached PDF