Week 6

6.1 a)

Python-line pseudocode

```
def buble_sort(arr):
    size = len(arr) # arr size
    for i in range(size): # iters through whole array + compares each element & swaps if needed
      for j in range(0, n-i-1):
          if arr[j] > arr[j+1]: # if the righter element is smaller , swap them
              swap(arr[j], arr[j+1])
```

b)

worst case, most swaps. When you have to perform a swap during every nested iter causing n · n swaps = $O(n^2)$

Average case, most elements are sorted but a comparisson of all elements must be performed so the time complexity is $\Theta(n^2)$

Best case is when the array is sorted so you iter the array once resulting in a time complexity of $\Omega(n)$