NP HW

1. Algorithm: Run bubble sort once from left to right we will have the biggest number to the right from the array. and then we will do Bubble sort reversely to locate the smallest number from the array

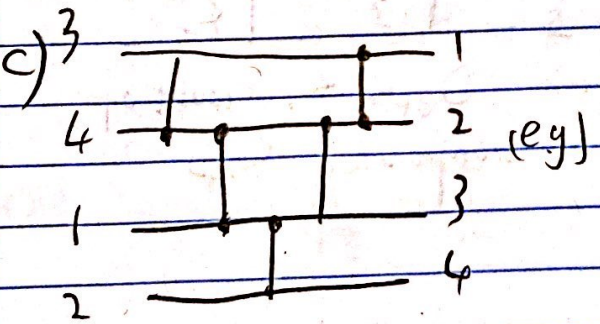
a)

Sudo code:

```
for(i<0 to (n-1)){
  if (A[i] > A[i+1]){
    swap([Ai], A[i+1])
  }
}
for(i<n-1 to 0){
  if (A[i] < A[i-1]){
    swap(A[i], A[i-1])
  }
}
```

b) Size: $(n-1)+(n-2) = 2n-3 \Rightarrow O(2n-3) \Rightarrow O(n)$

c)




(eg)

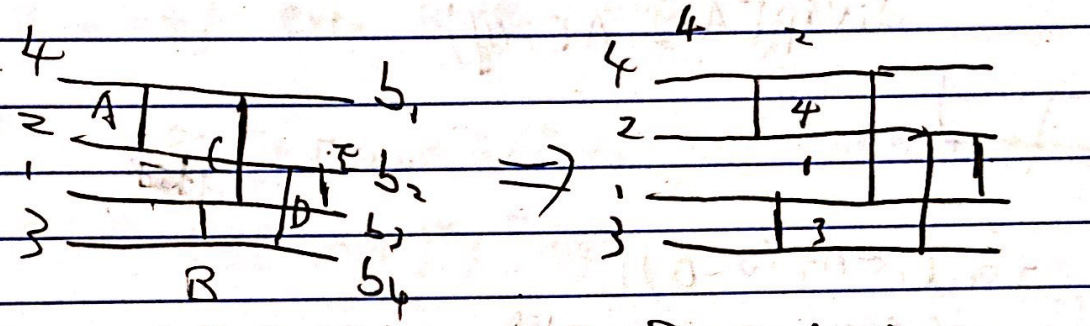
Depth = 4

Comparison

2 a) If sorting solved on a ~~merging~~ network, depth $O(\log(n))$, half sorting we have the size

So $\log(n)$. become $\log(\frac{n}{2})$ in big O notation, this will be represented as $O(\log n)$ coefficient will not be considered in big O

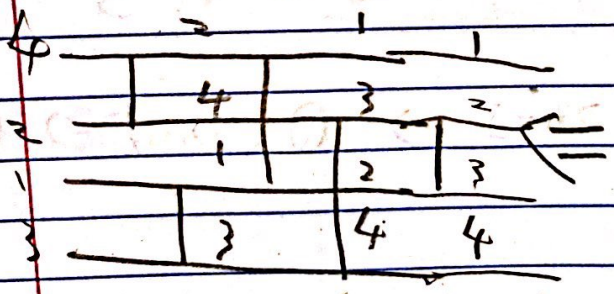
b)



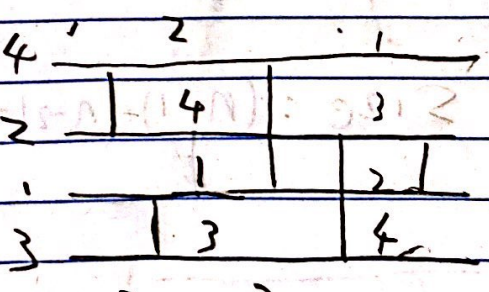

Input (4, 2, 1, 3)
When time is 0,




Depth 1, result of comparator A, B




Depth 3. result of comparator E




Depth 2. result of comparator C, D

3. Yes. Sorting is PL complete, therefore Half sorting PL should be complete as well. If half sorting is not PL-complete, Koco and Nevel said PL complete. Based on question 2 we can say sorting can be solved by half sorting