Algorithms 1D

1 a 8 3 4 6 5 7 2

Bubbling left to right

1st pass	3	4	6	5	7	2	8
2nd pass	3	4	5	6	2	7	8
3rd pass	3	4	5	2	6	7	8
4th pass	3	4	2	5	6	7	8
5th pass	3	2	4	5	6	7	8
6th pass	2	3	4	5	6	7	8

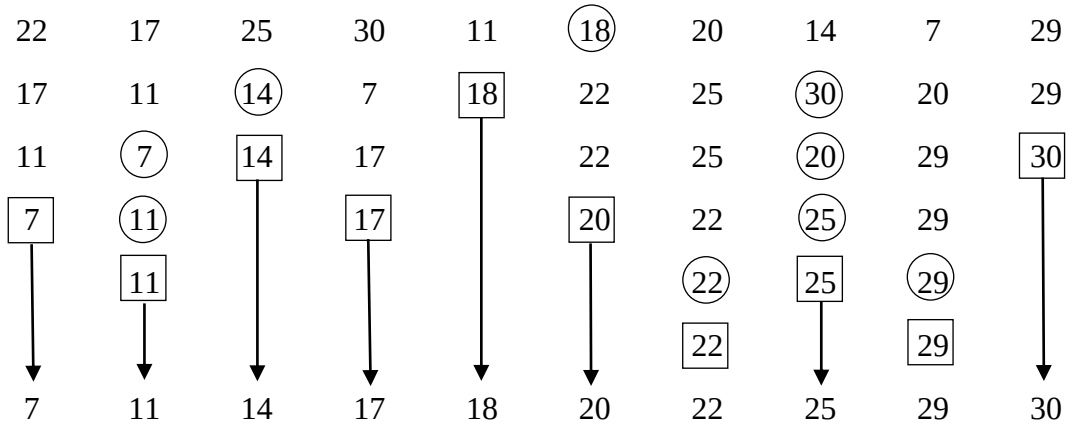
sort complete

b Bubbling left to right

1st pass	8	4	6	5	7	3	2
2nd pass	8	6	5	7	4	3	2
3rd pass	8	6	7	5	4	3	2
4th pass	8	7	6	5	4	3	2

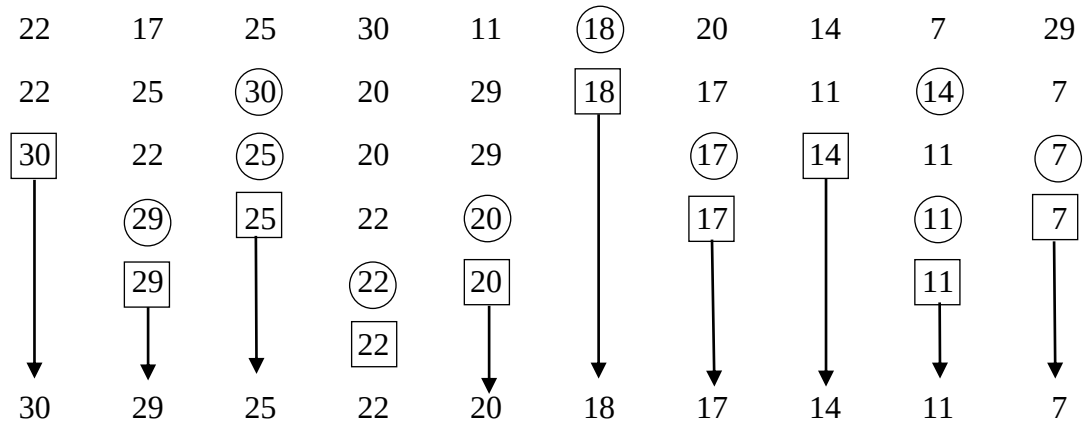
sort complete

2 a



sort complete

2 b



sort complete

3 a

N H R K S C J E M P L

Bubbling from left to right.

H N K R C J E M P L S

H K N C J E M P L R S

H K C J E M N L P R S

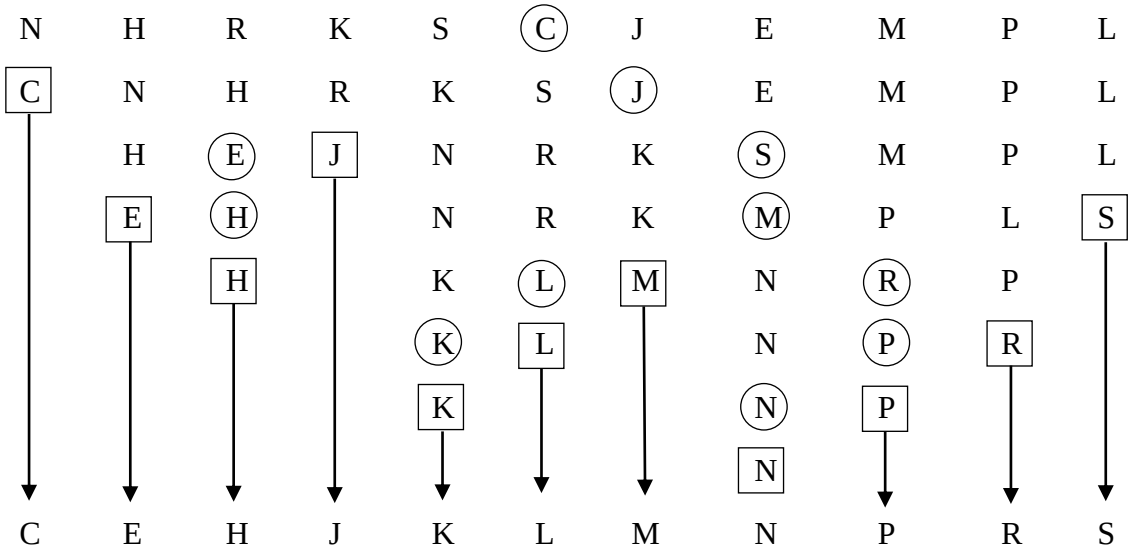
H C J E K M L N P R S

C H E J K L M N P R S

C E H J K L M N P R S

sort complete

3 b



sort complete

4 a

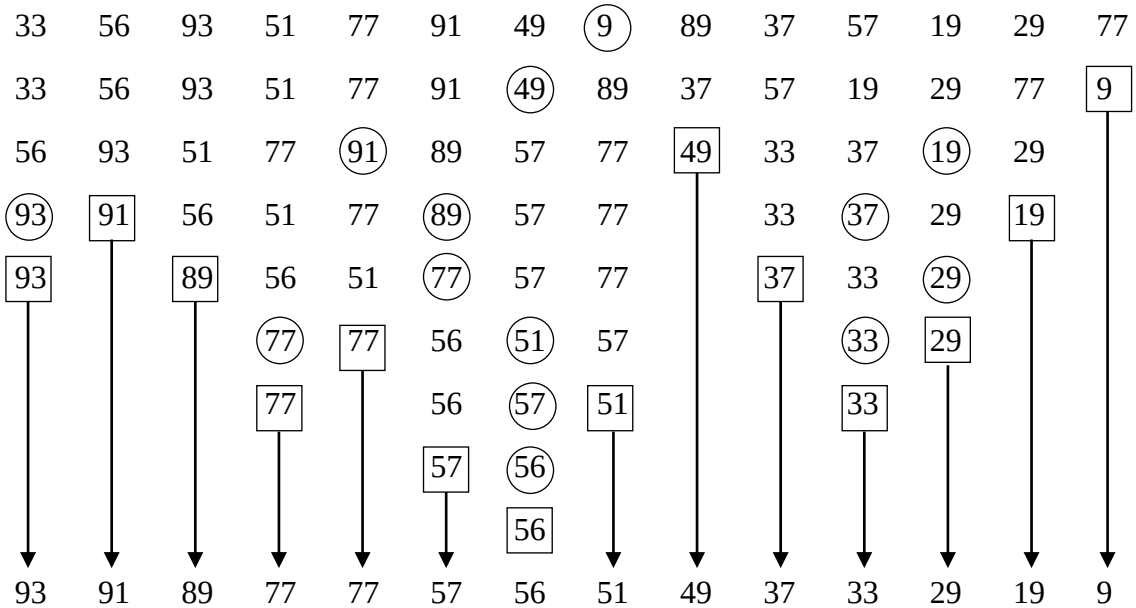
33	56	93	51	77	91	49	9	89	37	57	19	29	77
56	93	51	77	91	49	33	89	37	57	19	29	77	9
93	56	77	91	51	49	89	37	57	33	29	77	19	9
93	77	91	56	51	89	49	57	37	33	77	29	19	9
93	91	77	56	89	51	57	49	37	77	33	29	19	9
93	91	77	89	56	57	51	49	77	37	33	29	19	9
93	91	89	77	57	56	51	77	49	37	33	29	19	9
93	91	89	77	57	56	77	51	49	37	33	29	19	9
93	91	89	77	57	77	56	51	49	37	33	29	19	9
93	91	89	77	77	57	56	51	49	37	33	29	19	9

sort complete

So list is:

Amy	93	Annie	51
Greg	91	Harry	49
Janelle	89	Josh	37
Sophie	77	Alex	33
Dom	77	Sam	29
Lucy	57	Myles	19
Alison	56	Hugo	9

4 b



5 a For n items the maximum number of comparisons made is given by

$(n-1)+(n-2)+...3+2+1=\frac{n(n-1)}{2}$ This occurs when the initial list is in descending order.

b The bubble sort would be quicker, for example, if the items are to be put in increasing order and if the only item out of place is the largest.

c i Bubble sort. Only the 7 is out of place and it will be moved to its final position in the first pass. A second pass is still needed to complete the bubble sort. A total of 11 comparisons is needed for the bubble sort and 14 are needed for the quick sort.

ii Quick sort. This is the worst case for the bubble sort. The 1 is at the wrong end of the list and only moves one place with each pass.

6 a There are 9 names in the list, so the middle will be $\frac{9+1}{2}=5$

The pivot is the 5th name (Mellor) on the list and taken as the pivot. Starting at the beginning of the list, each name is compared with Mellor and placed on the left side if it comes before Me or the right side if it comes after Me to produce two sub-lists. The process is repeated for each sub-list with pivot of G on the left and Mi on the right. Select further pivots from within each sub-list and repeat the process.

b

H	S	F	G	(Me)	C	Mi	W	A	(M is the pivot)
H	F	(G)	C	A	[Me]	S	(Mi)	W	(1st pass)
F	(C)	A	[G]	(H)	[Me]	[Mi]	S	(W)	(2nd pass)

1st pass: pivots are G and Mi
2nd pass: pivots are C, H and W

Challenge

a Answers will vary.

b When the cards are in order, we have A♥, 2♥... K♥, A♣ ... K♣, A♦ ... K♦, A♠ ... K♠.
Recall that bubble sort compares two cards at a time, swaps them if they are not in order and leaves the pair without change if they are in the required order. We are asked to make a single change to the arrangement above so that bubble sort requires 51 passes to put the cards back in order. In other words, we want the algorithm to go through the whole deck 51 times, each time making at least one swap, before the cards are back in order. This can be achieved by putting the A♥ at the very bottom of the deck. Then at each pass we make exactly one swap, interchanging the positions of A♥ and the card directly before it. The rest of the deck remains unchanged.
For example:
Pass 1: 2♥... K♥, A♣ ... K♣, A♦ ... K♦, A♠ ... K♠, A♥
Changes into: 2♥... K♥, A♣ ... K♣, A♦ ... K♦, A♠ ... A♥, K♠
Pass 2: 2♥... K♥, A♣ ... K♣, A♦ ... K♦, A♠ ... Q♠, A♥, K♠
Changes into: 2♥... K♥, A♣ ... K♣, A♦ ... K♦, A♠ ... A♥, Q♠, K♠
Since we need to move the ace ahead of 51 cards, the algorithm will take exactly 51 passes to put the cards back in order.