

Topic 4 · Sheet A: Questions

Bubblesort: correctness and running time

Suggested time

About 75 minutes

Total marks

73

Questions

12

Builds on

Topics 1 to 3

How to use this sheet. Work on your own paper. The questions start gently and get harder; questions 11 and 12 are exam style. Marks are a guide to how much working is expected, roughly one minute per mark. You need Bubblesort, its loop invariant, and how to count its comparisons and swaps.

Bubblesort. Unless a question says otherwise, “Bubblesort” means this algorithm. An *inversion* of A is a pair of positions $p < q$ with $A[p] > A[q]$.

```

1  for  $i = 1, \dots, n - 1$  do
2      for  $j = n$  down to  $i + 1$  do
3          if  $A[j - 1] > A[j]$  then swap  $A[j - 1]$  and  $A[j]$ 

```

1

4 marks

Run the first outer iteration ($i = 1$) of Bubblesort on $A = [4, 2, 5, 1, 3]$.

- (a) Write down the array after this iteration. [2]
- (b) How many comparisons were made in this iteration? [1]
- (c) How many swaps were made in this iteration? [1]

2

4 marks

Continue the run of Bubblesort from question 1.

- (a) Write down the array after the iteration $i = 2$. [1]
- (b) Write down the array after the iteration $i = 3$. [1]
- (c) How many comparisons does the whole run make? [1]
- (d) How many swaps does the whole run make? [1]

3

6 marks

- (a) State the invariant that holds at the start of iteration i of the outer loop. [2]
- (b) Check the invariant on your array from question 2(a), at the start of iteration $i = 3$. [2]
- (c) Explain why the invariant, applied at the end of the last iteration, shows that the array is sorted. [2]

4

6 marks

- (a) How many comparisons does iteration i of the outer loop make? [1]
- (b) Find the total number of comparisons on an array of size n . [2]
- (c) Give the running time of Bubblesort in Θ notation. [1]
- (d) How many comparisons does Bubblesort make when $n = 10$, and when $n = 100$? [2]

5

6 marks

- (a) How many swaps does Bubblesort make on an array that is already sorted in increasing order? [1]
- (b) How many swaps does it make on an array sorted in decreasing order? [2]
- (c) Explain why the running time is still $\Theta(n^2)$ on an array that is already sorted. [1]
- (d) Explain why swapping two adjacent entries with $A[j - 1] > A[j]$ reduces the number of inversions by exactly 1. Deduce that the total number of swaps equals the number of inversions in the input. [2]

6

6 marks

Consider the first iteration ($i = 1$) of Bubblesort on an array of distinct numbers.

- (a) Explain why the smallest entry ends up in position 1. [2]
- (b) If the smallest entry starts at position k , how many of the swaps in this iteration involve it? [2]
- (c) Explain why iteration i never changes $A[1], \dots, A[i - 1]$. [2]

7

6 marks

In a variant, a flag records whether an outer iteration made any swap. If an iteration makes no swap, the algorithm stops at once.

- (a) Explain why it is correct to stop after an iteration with no swaps. [2]
- (b) How many comparisons does the variant make on an array that is already sorted? [2]
- (c) What is the variant's worst-case running time? Give an input that achieves it. [2]

8

6 marks

Another version moves large entries to the right instead.

```

1  for  $i = 1, \dots, n - 1$  do
2      for  $j = 1, \dots, n - i$  do
3          if  $A[j] > A[j + 1]$  then swap  $A[j]$  and  $A[j + 1]$ 

```

- (a) Run the first outer iteration on $A = [3, 1, 4, 2]$ and write down the array afterwards. [2]
- (b) State an invariant for this version, at the start of iteration i . [2]
- (c) How many comparisons does this version make in total? [2]

9

6 marks

A naive sorting method tries every ordering of the n distinct entries until it finds the sorted one.

- (a) How many orderings are there? [1]
- (b) Checking whether one ordering is sorted takes $n - 1$ comparisons. Give the worst-case number of comparisons. [2]
- (c) Compare this with Bubblesort when $n = 10$. [2]
- (d) Which of $n!$ and n^2 is ω of the other? [1]

10

7 marks

Bubblesort also works when entries repeat. Let $A = [2, 1, 2, 1]$.

- (a) Run Bubblesort on A , giving the array after each outer iteration. [3]
- (b) How many swaps did it make? Check this against the number of inversions. [2]
- (c) Explain why two equal entries never change their order relative to each other. [2]

11

8 marks

Exam style. Let $A = [5, 3, 8, 1, 9, 2]$.

- (a) Give the array after each of the first two outer iterations of Bubblesort. [3]
- (b) How many comparisons does the whole run make? [1]
- (c) Count the inversions of A , and hence state how many swaps the whole run makes. [2]
- (d) After the first two iterations, which entries are certain to be in their final positions? Explain. [2]

12

8 marks

Exam style. Use the variant from question 7, which stops after an iteration with no swaps. Let $n \geq 3$, and consider the two nearly sorted arrays

$$P = [2, 3, \dots, n, 1] \quad \text{and} \quad Q = [n, 1, 2, \dots, n - 1].$$

- (a) How many swaps are needed to sort each array? [2]

- (b) How many comparisons does the variant make on P ? [2]
- (c) How many comparisons does the variant make on Q ? [2]
- (d) Explain why the two arrays behave so differently. [2]

End of questions. Worked answers and the mark scheme are in the matching answers PDF.