Solutions for Exercise Sheet 6

Handout: Oct 14th — Deadline: Oct 23rd - 4pm

Question 6.1 (marks 0.5)

Consider the following algorithm. Does it sort correctly? (You might want to work out your own example to understand this better.)

DO-I-SORT(A, n)

```
1: for i = 1 to n do
2:     for j = 1 to n do
3:         if A[i] < A[j] then
4:             exchange A[i] with A[j]
```

1. If the algorithm is correct prove its correctness by loop invariant. Otherwise argue why it is not correct eg., provide an instance where it fails.

   Solution: the loop invariant is: *at the end of iteration i of the outer loop, the elements in $A[1]..A[i]$ are in sorted order and $A[i]$ contains the largest element in $A[1], \ldots, A[n]$.*

   Initialisation: At initialisation the elements in $A[1]..A[1]$ are in sorted order and by the end of the first iteration $A[1]$ contains the largest element of the array.

   Maintenance: Let $A[1]..A[i-1]$ be sorted at the beginning of the iteration due to the loop invariant and $A[i-1]$ is the maximum of the whole array. Consider first the elements from $j = 1$ to $j = i$. The ones that are smaller than $A[i]$ are untouched. Once an element larger than $A[i]$ is found, then all the elements up to $A[i]$ are swapped (because they are sorted by loop invariant) except for equal ones re-establishing the loop invariant. Once $j = i + 1$ all elements are left untouched because $A[i]$ is already the maximum.

   Termination: at the end of the last iteration of the outer loop i.e., $i = n$, the loop invariant states that the array is sorted. then j is incremented to $j + 1$ and the algorithm terminates with the correct output.

2. State the runtime of the algorithm in asymptotic notation. Justify your answer.

   Solution: There are two nested for loops both going from 1 to n so the runtime is $\Theta(n^2)$ for all inputs.

Question 6.2 (0.5 marks)

Consider the following input for RANDOMIZED-QUICKSORT:

12	10	4	2	9	6	5	25	8

What is the probability that:

1. The elements $A[2] = 10$ and $A[3] = 4$ are compared?
2. The elements $A[1] = 12$ and $A[8] = 25$ are compared?
3. The elements $A[4] = 2$ and $A[8] = 25$ are compared?
4. The elements $A[2] = 10$ and $A[7] = 5$ are compared?

Solution: The elements in increasing order are as follows:

$$z_1 = 2; z_2 = 4; z_3 = 5; z_4 = 6; z_5 = 8; z_6 = 9; z_7 = 10; z_8 = 12, z_9 = 25$$

So the probabilities are:

1. $A[2] = 10$ and $A[3] = 4 \iff P(z_7 \text{ and } z_2) = \frac{2}{j-i+1} = \frac{2}{7-2+1} = \frac{1}{3}$
2. $A[1] = 12$ and $A[8] = 25 \iff P(z_8 \text{ and } z_9) = \frac{2}{j-i+1} = \frac{2}{2-1+1} = 1$
3. $A[4] = 2$ and $A[8] = 25 \iff P(z_1 \text{ and } z_9) = \frac{2}{j-i+1} = \frac{2}{9-1+1} = \frac{2}{9}$
4. $A[2] = 10$ and $A[7] = 5 \iff P(z_7 \text{ and } z_3) = \frac{2}{j-i+1} = \frac{2}{7-3+1} = \frac{2}{5}$

Question 6.3 (1 mark)

Prove that the expected runtime of RANDOMIZED-QUICKSORT is $\Omega(n \log n)$.

(HINT: It may be useful to consider how long it takes to compare $n/2$ elements to achieve a lower bound on the runtime.)

Solution: The expected runtime of the algorithm is (from lecture)

$$E[X] = 2\sum_{i=1}^{n-1}\sum_{k=1}^{n-i}\frac{1}{k+1} > 2\sum_{i=1}^{n/2}\sum_{k=1}^{n-i}\frac{1}{k+1} \tag{1}$$

where on the right side we only sum the first $n/2$ terms. Each term of the sum is greater than:

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n/2} = \sum_{k=1}^{n/2}\frac{1}{k+1} = \left(\sum_{k=1}^{n/2}\frac{1}{k}\right) - 1 \geq \ln(n/2) - 1 = \ln n - \ln 2 - 1$$

Plugging this into Eq (1):

$$E[X] = 2\sum_{i=1}^{n-1}\sum_{k=1}^{n-i}\frac{1}{k+1} > 2\sum_{i=1}^{n/2}\sum_{k=1}^{n-i}\frac{1}{k+1} \geq 2\sum_{i=1}^{n/2}(\ln n - \ln 2 - 1) = 2\frac{n}{2}(\ln n - \ln 2 - 1)$$

$$\geq n \ln n - 2n = \Omega(n \log n)$$

Question 6.4 (1 mark)

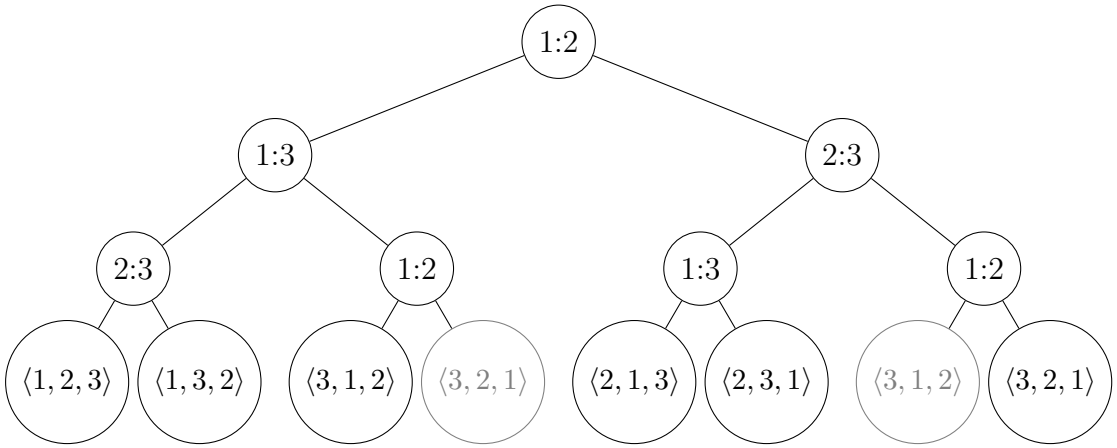
Draw the decision tree that reflects how SELECTIONSORT sorts $n = 3$ elements. Assume that all elements are mutually distinct.

For convenience here's the pseudocode again:

SELECTION-SORT(A)

```
1: n = A.length
2: for j = 1 to n − 1 do
3:     smallest = j
4:     for i = j + 1 to n do
5:         if A[i] < A[smallest] then smallest = i
6:     exchange A[j] with A[smallest]
```

Solution: In the following tree, edges going left represent the outcome "$\leq$" and edges going right represent the outcome ">".



The leaves drawn in gray are never actually reached as the same comparison was made earlier and we wouldn't have come down the tree this way.

Question 6.5 (0.5 marks)

What is the smallest possible depth of a leaf in a decision tree for a comparison sort?

Solution: For a permutation $a_1 \leq a_2 \leq \ldots a_n$ there are $n-1$ pairs of relative sorting, thus the smallest possible depth is $n-1$.

Eg., $n = 3$: $(a_1 <= a_2), (a_2 <= a_3) =>< 1, 2, 3 >$, and depth is 2.

Question 6.6 (0.25 marks)

Implement RANDOMIZED-QUICKSORT and solve the 'Yet Another Quicksort' Problem.