Assignment 5

Radityo Eko Prasojo, Werner Nutt,
Rafael Penaloza, Guohui Xiao

Mysteries and Other Array Algorithms

1. A Mystery Procedure

Consider the following procedure that takes as input an array and two indices.

```
1:  procedure MYSTERY(A, l, r)
2:      range := r − l + 1
3:      subrange := ⌈2 · range/3⌉
4:      if range = 2 and A[l] > A[r] then
5:          SWAP(A, l, r)
6:      else if range ≥ 3 then
7:          MYSTERY(A, l, l + subrange − 1)
8:          MYSTERY(A, r − (subrange − 1), r)
9:          MYSTERY(A, l, l + subrange − 1)
10:     end if
11: end procedure
```

Note that division in line 3 is division of real numbers and recall that the ceiling function $\lceil x \rceil$ returns the least integer that is greater or equal to x.

1. What effect does the call MYSTERY$(A, 1, A.\text{length})$ have on an array A? Explain your answer.

2. What is the asymptotic running time of MYSTERY?
   Hint: Use an appropriate technique from the lecture.

(6 Points)

2. Matching Pairs

Let s be an integer and A an array of integers. Then A has a *matching pair* for s if there are two distinct positions i, j such that $A[i] + A[j] = s$. The matching pair problem is to check, given s, whether A has a matching pair for s.

1. Develop an efficient algorithm in pseudocode that solves the matching pair problem.
   Hint: An algorithm that you know from the lecture may be useful.

2. What is the asymptotic worst-case complexity of your algorithm? Explain your answer.

3. Explain why your algorithm is correct. That is, show that whenever your algorithm returns `true`, there are two values in A that add up to s, and that if your algorithm returns `false`, there are no such values.
   Hint: Choose the right loop invariant.

(8 Points)

3. Duplicate Elimination without Order Preservation

Develop an algorithm that eliminates duplicate occurrences from an array. More precisely, the algorithm should take as input an array of integers A and return an array B that contains the same values as A, possibly in a different order, such that no value in B occurs more than once.

1. Write pseudocode for a straightforward solution, which need not be efficient. Explain the idea of that algorithm.

2. Write pseudocode for an efficient algorithm and explain the idea of that algorithm, too.
   Hint: Again, an algorithm that you know from the lecture may be useful.

3. What is the asymptotic worst-case complexity of each of the two algorithms? Explain your answer.

(8 Points)

4. Duplicate Elimination with Order Preservation

Develop now an algorithm that eliminates duplicate occurrences from an array of integers in such a way that the first occurrence of each value remains while all the later ones are dropped. For example, if the input array is

$$A = [3, 5, 9, 4, 5, 11, 9, 3, 1, 1, 3, 4, 5, 7],$$

then the output array is

$$B = [3, 5, 9, 4, 11, 1, 7].$$

Your goal is to develop an *efficient* algorithm. Implement your algorithm in the class `DuplicateEliminationWithOrderPreservation`, which you find in the zip file `DSA_A5.zip`.

1. Develop JUnit tests for the algorithm, covering both special and general cases. Include one test for the example above.
2. Write pseudocode for an efficient algorithm and explain the idea of that algorithm.
   Hint: The algorithm for Question 3 may come in handy.
3. What is the asymptotic worst-case complexity of the algorithm? Explain your answer.
4. Implement your algorithm as a method `eliminate`.

(8 Points)

Deliverables.

1. For question 4, hand in the Java file that you wrote.
2. Write one report for all tasks.

Combine all deliverables into one zip file, which you submit via the OLE website of the course. Please, follow the "Instructions for Submitting Course Work" on the Web page with the assignments, when preparing your coursework.
Submission: Until Mon, 18 April 2016, 23:55 hrs, to the OLE submission page of

Lab A / Lab B / Lab C