DIT181 Data Structures and Algorithms: Assignment 1

This document provides the assignment for weeks 2 and 3 of the DIT181 course. The assignment should be handed in by 2 February 2018. A test suite together with instructions on how to run it will be published separately.

Dynamic arrays

For the exercises involving code, please download the `Array2.java` skeleton file. The Array class implements a dynamic array using a fixed-size buffer. The array cannot grow beyond the size of the underlying buffer (`max_elements`).

Question 1

Implement the method `void reverse(int i, int x)`, which should reverse the array in place.

Determine the time complexity class of `reverse()`.

Question 2

Implement the method `int maxOdd()`, which should return the maximum odd number from the array, or `0` if the array has no odd elements.

Note Returning `0` to denote a special case is not ideal, but Java does not allow us to easily return special values.

Determine the time complexity class of `maxOdd()`.

Question 3

Suppose that you are going to implement removing an element of a given index from the array. What will be the time complexity of this operation? Implement it as the method `remove()`. What if you were allowed to change the order of the remaining elements? Implement as the method `remove2()`, which should be more efficient.

Question 4

Implement the method `int find(int x)`, which should return the index of the first occurrence of `x` in the array, or `-1` if `x` does not occur.

Determine the time complexity class of `find()`.

Question 5

(This question does not require implementation) Suppose that you have a dynamic array that expands by doubling the amount of elements, and that also allows for the elements to be removed from its end. To save space, you want to allow the array to shrink if at least half of the elements are free. Unfortunately, this proves to be inefficient. Explain what is the reason for inefficiency, and propose a better strategy.

CHALMERS TEKNISKA HÖGSKOLA AB
GÖTEBORGS UNIVERSITET
Data- och informationsteknik
412 96 Göteborg
Telefon: 031-772 10 00 (CTH) / 031-786 00 00 (GU)

Question 6

Implement the method `int maxPalindrome()`, which should Find the length of the longest palindrome that is a contiguous subsequence of the array. A palindrome is a word of the form `ABCBA` (length 5) or `ABCCBA` (length 6). Come up with the best solution you can in reasonable time.

Determine the time complexity class of `maxPalindrome()`.

Complexity

Question 7

Determine the big-O complexity classes of the following functions:

- $\sum_{i=1}^{n} i$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} 2^i$
- $\sum_{i=1}^{\log n} 2^n$
- $\sum_{i=1}^{n} 1/i$

Question 8

Show that $4^n \notin O(2^n)$.

Question 9

Consider the problem of evaluating a single variable polynomial of degree n:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} a_n x^n$$

The value of the polynomial for a particular value of x, can be computed by performing some sequence of additions and multiplications. A naïve algorithm can perform it using $O(n^2)$, operations (an operation is either an addition or a multiplication). There are at least two ways to improve it: one where you speed up the computation of x^n, while keeping the formula the same, and another one where you use a different formula. Come up with two improved algorithms.

Recurrences, divide and conquer

Question 10

Consider the following code:

```
import java.math.BigInteger;
public static BigInteger fib(int n) {
    if (n <= 1) return BigInteger.ONE;
    return fib(n-1).add(fib(n-2));
}
```

What is the complexity of this function assuming that each addition is $O(1)$?

Hint: What is the relationship between the run time and the result returned by the function?

Propose a faster implementation of the same function, and give its complexity.

CHALMERS TEKNISKA HÖGSKOLA AB
GÖTEBORGS UNIVERSITET
Data- och informationsteknik
412 96 Göteborg
Telefon: 031-772 10 00 (CTH) / 031-786 00 00 (GU)

Question 11

Solve the following recurrences:

- $T(n) = T(n-1) + \Theta(1/n)$
- $T(n) = 3T(n/2) + \Theta(n^2)$
- $T(n) = 4T(n/2) + \Theta(n^2)$
- $T(n) = 5T(n/2) + \Theta(n^2)$

Question 12

Implement maximum-subarray algorithm from Lecture 4 as the method `maxInterval()`. Please note that you will need to define an auxiliary method.

Question 13

Implement the method `findSplice()`, which expects the array to contain a sequence of integers that is a cyclic shift of a non-decreasing sequence. For example, here is such a sequence.

$$[4, 6, 7, 9, 0, 1, 3]$$

The method should return the index of the smallest element in the array. Use the divide and conquer technique, and propose a $O(\lg n)$ algorithm.

Question 14

Implement the method `median()`, which should return the median element of the array. Use the divide and conquer technique, and the method `partition()`. The method should have the $O(n)$ time complexity in the average case, assuming that the order of elements in the array is random.

Sorting

Question 15

Solving this question will require you to work with the `Lab1Sorting.java` skeleton file. You should implement three sorting algorithms: insertion sort, Quicksort and merge sort. Insertion sort and Quicksort should sort the array in-place, but not merge sort.