DIT181 Data Structures and Algorithms: Exercises 1

This document provides a set of exercises for week 2 of the DIT181 course. Unlike the assignments, the exercises are not to be handed in.

Dynamic arrays

For the exercises involving code, please download the `Array.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`).

Exercise 1

Implement the method `void insert(int i, int x)` so that it inserts element `x` into the array at index `i`. If the index is equal to the array size, the new element is appended. Otherwise, if the index is smaller, some elements of the array need to be shifted.

Decide in what complexity class is the time complexity of `insert()`. Express the complexity using the $O()$ notation, or possibly using the $\theta()$ notation, and using n as the size of the array, and i as the index.

Exercise 2

Implement the method `boolean isSorted()` method, which should return `true` if the array is sorted, and `false` otherwise.

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

Exercise 3

Implement the method `int find(int x)` method, 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()`.

Exercise 4

Implement the method `int maxNonDecreasing()`, which should the length of the longest non-decreasing contiguous subsequence of the array.

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

Exercise 5

Implement the method `int subArrayIndex(Array b)`, which should find the index of the first subarray of the array that is equal to array b. If such a subarray does not exist, it should return -1. Come up with the best solution you can in reasonable time.

CHALMERS TEKNISKA HÖGSKOLA AB

GÖTEBORGS UNIVERSITET

Data- och informationsteknik

412 96 Göteborg

Telefon: 031-772 10 00 (CTH) / 031-786 00 00 (GU)

Determine the time complexity class of `subArrayIndex()`. Is the ‘reasonable’-case complexity better than the worst-case complexity? (Define the ‘reasonable’ case).

Complexity

Exercise 6

Assume that your computer can perform about 10^9 operations per second. Estimate how large inputs can be processed in different times using algorithms from different complexity classes, by filling in the following table:

	1 second	**1 minute**	**1 hour**	**1 day**	**1 month**	**1 year**	**1 century**
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
$n^2 \lg n$							
n^3							
2^n							
$n!$							

Exercise 7

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

- $2x^2 + 16x$
- $3x \log x + x + \frac{x^2}{13 \log x}$
- $2^x + 10^x$
- $(x - \lg x)\left(x - 2\sqrt{x}\right) + 4x \lg x$
-

Exercise 8

Prove that $\log_2 n \in \Theta(\log_{10} n)$

Note $\log_2 n$ is often written as $\lg n$, whereas $\log_{10} n$ is often written as $\log n$.

Exercise 9

Show that $\lg n \in O(\sqrt{n})$, but not $\sqrt{n} \in O(\lg n)$

CHALMERS TEKNISKA HÖGSKOLA AB

GÖTEBORGS UNIVERSITET

Data- och informationsteknik

412 96 Göteborg

Telefon: 031-772 10 00 (CTH) / 031-786 00 00 (GU)

Exercise 10 (optional)

This exercise is recommended only for interested students familiar with Python (please note that the rest of the course, including the exam, will all be using Java). Consider the Python program from lecture 2 that read a file into a buffer character by character.

```
import sys
s = ''
num_chars = 0
while True:
  c = sys.stdin.read(1)
  s2 = s
  if c == '':
    break
  s += c
  num_chars += 1
print num_chars
print s,
```

This version of the program was naïve and inefficient, yielding quadratic behaviour. If we make a small change and remove the line `s2 = s`, however, the program will become quite efficient. The quadratic behaviour will be gone due to a particular optimisation performed by the runtime of Python (`cpython`). Your job is to search around and find out what optimisation is used. Please do not spend a lot of time on this question.

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