Last name(s) **Name** **ID**

Midterm EDA exam Length: 2.5 hours 07/11/2016

- *The exam has 4 sheets, 8 sides and 4 problems.*
- *Write your full name and ID on every sheet.*
- *Write your answers to all problems in the exam sheets within the reserved space.*
- *Unless otherwise indicated, all your answers must be justified.*

Problem 1 (2 points)

In this problem you do not need to justify your answers.

(a) (0.8 pts.) Fill the gaps in the following table (except for the cell marked with **Do not fill**) with the costs in time for sorting a vector of integers of size n using the indicated algorithms. Assume uniform probability in the average case.

	Best case	*Average case*	*Worst case*
Quicksort (with Hoare's partition)			
Mergesort			
Insertion		**Do not fill**	

(b) (0.2 pts.) The solution to the recurrence $T(n) = 2T(n/4) + \Theta(1)$ is

(c) (0.2 pts.) The solution to the recurrence $T(n) = 2T(n/4) + \Theta(\sqrt{n})$ is

(d) (0.2 pts.) The solution to the recurrence $T(n) = 2T(n/4) + \Theta(n)$ is

(e) (0.3 pts.) What does Karatsuba's algorithm compute? What is its cost?

(it continues on the back)

(f) (0.3 pts.) What does Strassen's algorithm compute? What is its cost?

Last name(s) **Name** **ID**

Problem 2 **(3 points)**

Given $n \geq 2$, we say that a sequence of n integers $a_0, \ldots, a_{n-1}$ is *bi-increasing* if $a_{n-1} < a_0$ and there exists an index t (with $0 \leq t < n$) that satisfies the following conditions:

- $a_0 \leq \ldots \leq a_{t-1} \leq a_t$
- $a_{t+1} \leq a_{t+2} \leq \ldots \leq a_{n-1}$

For example, the sequence $12, 12, 15, 20, 1, 3, 3, 5, 9$ is bi-increasing (take $t = 3$).

(a) (2 pts.) Implement in C++ a function

bool *search* (**const** *vector* <**int**>& *a*, **int** *x*);

which, given a vector a that contains a bi-increasing sequence and an integer x, returns whether x appears in the sequence or not. If you use auxiliary functions that are not part of the C++ standard library, implement them too. The solution must have cost $\Theta(\log(n))$ in time in the worst case.

(it continues on the back)

(b) (1 pt.) Justify that the cost in time in the worst case of your function *search* is $\Theta(\log(n))$. When does this worst case take place?

Last name(s) **Name** **ID**

Problem 3 **(2 points)**

Consider the following function:

```
int mystery(int m, int n) {
  int p = 1;
  int x = m;
  int y = n;
  while (y ≠ 0) {
    if (y % 2 == 0) {
      x *= x;
      y /= 2;
    }
    else {
      y −= 1;
      p *= x;
  } }
  return p;
}
```

(a) (1 pt.) Given two integers $m, n \geq 0$, what does *mystery*(m,n) compute? You do not need to justify your answer.

(b) (1 pt.) Analyse the cost in time in the worst case as a function of n of *mystery*(m, n).

This side would be intentionally blank if it were not for this note.

Last name(s) **Name** **ID**

Problem 4 **(3 points)**

The *Fibonacci sequence* is defined by the recurrence $F(0) = 0$, $F(1) = 1$ and

$$F(n) = F(n-1) + F(n-2) \quad \text{if } n \geq 2.$$

(a) (0.5 pts.) Let $\phi = \frac{\sqrt{5}+1}{2}$ be the so-called *golden number*. Prove that $\phi^{-1} = \phi - 1$.

(b) (1.5 pts.) Prove that, for any $n \geq 0$, we have

$$F(n) = \frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}$$

(it continues on the back)

(c) (1 pt.) Prove that $F(n) = \Theta(\phi^n)$.