Last name **Name** **ID**

Midterm EDA Exam Length: 2.5 hours 19/10/2015

- *The exam has 5 sheets, 10 sides and 3 problems.*
- *Write your full name and ID on every sheet.*
- *Write your answers within the reserved space.*
- *All your answers must be justified.*

Problem 1 (3.5 points)

Fibonacci numbers are defined by the recurrence $f_k = f_{k-1} + f_{k-2}$ for $k \geq 2$, with $f_0 = 0$ and $f_1 = 1$. Answer the following exercises:

(a) (0.5 points) Consider the following function *fib1* which, given a non-negative integer k, returns f_k:

```
int fib1(int k) {
  vector<int> f(k+1);
  f[0] = 0;
  f[1] = 1;
  for (int i = 2; i <= k; ++i)
    f[i] = f[i-1] + f[i-2];
  return f[k];
}
```

Describe the asymptotic cost *in time and space* of *fib1* (k) as a function of k as precisely as possible.

(b) (1 point) Show that, for $k \geq 2$, the following matrix identity holds:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{k-1} = \begin{pmatrix} f_k & f_{k-1} \\ f_{k-1} & f_{k-2} \end{pmatrix}$$

(c) (1 point) Fill the gaps in the following code so that function *fib2* (k) computes f_k, given a $k \geq 0$.

```
typedef vector<vector<int>> matrix;

matrix mult(const matrix& A, const matrix& B) {
  // Pre: A and B are square matrices of the same dimensions
  int n = A.size ();
  matrix C(n, vector<int>(n, 0));
  for (int i = 0; i < n; ++i)
    for (int j = 0; j < n; ++j)
      for (int k = 0; k < n; ++k)
        
  return C;
}
```

Last name **Name** **ID**

```
matrix mystery(const matrix& M, int q) {
  int s = M.size();
  if (q == 0) {
    matrix R(s, vector<int>(s, 0));
    for (int i = 0; i < s; ++i) ____________
    return R;
  }
  else {
    matrix P = mystery(M, q/2);
    if ( ____________ ) return mult(P, P);
    else return ____________ ;
} }

int fib2(int k) {
  if (k ≤ 1) return k;
  matrix M = { {1, 1}, {1, 0} };
  matrix P = mystery(M, k−1);
  return ____________ ;
}
```

(d) (1 point) Describe the asymptotic cost in time of *fib2* (k) as a function of k as precisely as possible.

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

Last name **Name** **ID**

Problem 2 **(3.25 points)**

Given a vector of integers v and an integer x, the function

```
int position (const vector<int>& v, int x) {
  int n = v.size ();
  for (int i = 0; i < n; ++i)
    if (v[i] == x)
      return i;
  return −1;
}
```

examines the $n = v.size()$ positions of v and returns the first one that contains x, or -1 if there is none.

(a) (0.75 points) Describe as a function of n the asymptotic cost in time of *position* in the best case as precisely as possible. When can this best case take place?

(b) (0.75 points) Describe as a function of n the asymptotic cost in time of *position* in the worst case as precisely as possible. When can this worst case take place?

(c) (0.75 points) Show that for any integer $n \geq 1$, the following identity holds:

$$\sum_{i=1}^{n} \frac{i}{2^i} = 2 - \frac{n}{2^n} - \frac{1}{2^{n-1}}.$$

Last name **Name** **ID**

(d) (1 point) Assume that we have a probability distribution over the input parameters. Namely, the probability that x is element $v[i]$ is $\frac{1}{2^{i+1}}$ for $0 \leq i < n-1$, and that it is element $v[n-1]$ is $\frac{1}{2^{n-1}}$. In particular, these probabilities add up 1, so that x is always one of the n values of v.

Describe as a function of n the asymptotic cost in time of *position* in the average case as precisely as possible.

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

Last name **Name** **ID**

Problem 3 **(3.25 points)**

In this exercise we tackle the problem of, given two positive integers a and b, to compute their greatest common divisor $\gcd(a,b)$. Recall that $\gcd(a,b)$ is, by definition, the only positive integer g such that:

1. $g \mid a$ (g divides a),
2. $g \mid b$,
3. if $d \mid a$ and $d \mid b$, then $d \mid g$.

(a) (1.25 points) Show that the following identities are true:

$$\gcd(a,b) = \begin{cases} 2\gcd(a/2, b/2) & \text{if } a, b \text{ are even} \\ \gcd(a, b/2) & \text{if } a \text{ is odd and } b \text{ is even} \\ \gcd((a-b)/2, b) & \text{if } a, b \text{ are odd and } a > b \end{cases}$$

Hint: you can use that, given two positive integers a and b such that $a > b$, we have $\gcd(a,b) = \gcd(a-b, b)$.

(b) (1 point) Write a function **int** gcd(**int** a, **int** b) in C++ which, by using divide and conquer and exercise (a), computes the greatest common divisor $\gcd(a, b)$ of two given positive integer numbers a, b.

Hint: you can also use that for any positive integer a, $\gcd(a, a) = a$.

(c) (1 point) Assuming that a and b are positive integers, each of which represented with a vector of n bits, describe the cost in time in the worst case of $gcd(a, b)$ as a function of n as precisely as possible. When can this worst case take place?

Assume that the cost of the following operations with integers of n bits: adding, subtracting, comparing, multiplying/dividing by 2 is $\Theta(n)$, and that computing the remainder modulo 2 takes time $\Theta(1)$.