Solution Midterm Exam EDA 31/03/2016

Proposed solution to problem 1

(a)

	Best case	*Average case*	*Worst case*
Quicksort (with Hoare's partition)	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n^2)$
Mergesort	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$
Insertion	$\Theta(n)$	**Do not fill**	$\Theta(n^2)$

(b) $\Theta(\sqrt{n})$

(c) $\Theta(\sqrt{n}\, \log n)$

(d) $\Theta(n)$

(e) Karatsuba's algorithm computes the product of two natural numbers of n bits in time $\Theta(n^{\log_2 3})$.

(f) Strassen's algorithm computes the product of two matrices of size $n \times n$ in time $\Theta(n^{\log_2 7})$.

Proposed solution to problem 2

(a) A possible solution:

```
#include <vector>

using namespace std;

bool dic_search (const vector<int>& a, int l, int r, int x) {
  if (l > r) return false;
  int m = (l+r)/2;
  if (a[m] < x) return dic_search (a, m+1, r,   x);
  if (a[m] > x) return dic_search (a,   l, m−1, x);
  return true;
}

bool search (const vector<int>& a, int l, int r, int x) {
  if (l+1 == r) return a[l] == x or a[r] == x;
  int m = (l+r)/2;
  if (a[m] ≥ a[l]) {
    if (a[l] ≤ x and x ≤ a[m]) return dic_search (a, l, m, x);
    else                        return search (a, m, r, x);
  }
  else {
    if (a[m] ≤ x and x ≤ a[r]) return dic_search (a, m, r, x);
    else                        return search (a, l, m, x);
} }
```

```
bool search(const vector<int>& a, int x) {
  return search(a, 0, a.size()−1, x);
}
```

(b) Let $C(n)$ be the cost of dealing with a vector of size n (be it with function *search* or with function *dic_search*) in the worst case (for example, when the element x does not appear in the sequence). Apart from operations of constant cost (arithmetic computations, comparisons and assignments between integers, vector accesses), exactly one call is made over a vector of size half of that of the input. Therefore, the cost is determined by the recurrence:

$$C(n) = C(n/2) + \Theta(1),$$

which, by the master theorem of divisive recurrences, has solution $C(n) = \Theta(\log(n))$.

Proposed solution to problem 3

(a) It computes m^n.

(b) The cost of the function is determined by the cost of the loop. Each iteration requires time $\Theta(1)$, since only arithmetic operations and integer assignments are performed. Therefore, the cost is proportional to the number of iterations. We observe that if y is even, then y is reduced to half its value. And if y is odd with $y > 1$, then at the next iteration the value $y - 1$ is considered, which is even, and then it is reduced to half its value. Hence if $n \geq 1$ the number of iterations is between $1 + \lfloor \log(n) \rfloor$ and $1 + 2\lfloor \log(n) \rfloor$. Altogether, the cost is $\Theta(\log(n))$.

Proposed solution to problem 4

(a) We have that $\phi - 1 = \frac{\sqrt{5}+1}{2} - 1 = \frac{\sqrt{5}-1}{2}$, and then

$$\phi \cdot (\phi - 1) = \frac{\sqrt{5}+1}{2} \cdot \frac{\sqrt{5}-1}{2} = \frac{(\sqrt{5}+1)(\sqrt{5}-1)}{4} = \frac{5-1}{4} = 1$$

Hence $\phi^{-1} = \phi - 1$.

(b) By induction over n:

- **Base case $n = 0$:** we have $F(0) = 0$, and

$$F(n) = \left. \frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}} \right|_{n=0} = \frac{1-1}{\sqrt{5}} = 0.$$

- **Base case $n = 1$:** we have $F(1) = 1$, and

$$F(n) = \left.\frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}\right|_{n=1} = \frac{\phi + \phi^{-1}}{\sqrt{5}} = \frac{\sqrt{5}}{\sqrt{5}} = 1.$$

- **Inductive case $n > 1$:** by induction hypothesis,

$$F(n) = F(n-2) + F(n-1) = \frac{\phi^{n-2} - (-\phi)^{-(n-2)}}{\sqrt{5}} + \frac{\phi^{n-1} - (-\phi)^{-(n-1)}}{\sqrt{5}} =$$

$$= \frac{\phi^{n-1}(\phi^{-1} + 1) - (-\phi)^{-(n-1)}(-\phi + 1)}{\sqrt{5}} = \frac{\phi^{n-1} \cdot \phi - (-\phi)^{-(n-1)}(-\phi^{-1})}{\sqrt{5}} =$$

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

(c) We have that

$$\lim_{n\to+\infty} \frac{F(n)}{\phi^n} = \lim_{n\to+\infty} \frac{\frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}}{\phi^n} = \lim_{n\to+\infty} \frac{1 - (\frac{-1}{\phi^2})^n}{\sqrt{5}} = \frac{1}{\sqrt{5}}$$

since $\phi > 1$, $\phi^2 > 1$ and $\lim_{n\to+\infty}(\frac{-1}{\phi^2})^n = 0$. So $F(n) = \Theta(\phi^n)$.