Solució de l'Examen Parcial EDA — 19/10/2015

Proposta de solució al problema 1

(a) Anàlisi de cost:

- *En temps:* el cost de les inicialitzacions és $\Theta(k)$, per la creació del vector f. Com que el bucle s'executa $\Theta(k)$ vegades i cada iteració costa temps $\Theta(1)$, la contribució al cost del bucle és $\Theta(k)$. En total el cost és doncs $\Theta(k)$.
- *En espai:* $\Theta(k)$, ja que el consum de memòria està dominat per la creació del vector f, de mida $k+1$.

(b) Per inducció.

- *Cas base:* $k = 2$. Aleshores efectivament

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{k-1} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} f_2 & f_1 \\ f_1 & f_0 \end{pmatrix} = \begin{pmatrix} f_k & f_{k-1} \\ f_{k-1} & f_{k-2} \end{pmatrix}.$$

- *Cas inductiu:* $k > 2$. Per hipòtesi d'inducció:

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

$$= \begin{pmatrix} f_k + f_{k-1} & f_k \\ f_{k-1} + f_{k-2} & f_{k-1} \end{pmatrix} = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}.$$

(c) Una possible solució:

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

matrix mult(const matrix& A, const matrix& B) {
  assert (A.size () == A[0].size ());
  assert (B.size () == B[0].size ());
  assert (A.size () == B.size ());
  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)
        C[i][j] += A[i][k] * B[k][j];
  return C;
}

matrix misteri (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) R[i][i] = 1;
    return R;
  }
  else {
```

```
    matrix P = misteri (M, q/2);
    if (q % 2 == 0) return mult(P, P);
    else            return mult(mult(P, P), M);
} }

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

(d) Primer analitzem el cost de *misteri* (M, k) en funció de k, que anomenarem $C(k)$. Observem que les crides a *mult* sempre es fan amb matrius 2×2, i per tant triguen temps constant. Per tant el cost no recursiu és constant i tenim que $C(k) = C(k/2) + \Theta(1)$, d'on aplicant el Teorema Mestre de Recurrències Divisores s'obté que $C(k) = \Theta(\log k)$.

Així doncs, el cost de *fib2* (k) és $C(k-1) + \Theta(1) = \Theta(\log k)$.

Proposta de solució al problema 2

(a) Un cas millor es dóna quan x és a la primera posició, és a dir, x és $v[0]$. Aleshores només s'entra un cop al bucle, i el cost total és $\Theta(1)$.

(b) Un cas pitjor es dóna quan x no apareix al vector v. Aleshores es fan $\Theta(n)$ iteracions del bucle, cadascuna de les quals triga temps $\Theta(1)$. El cost total és $\Theta(n)$.

(c) Per inducció.

- *Case base:* $n = 1$. Tenim $\sum_{i=1}^{n} \frac{i}{2^i} = \frac{1}{2}$, i $2 - \frac{n}{2^n} - \frac{1}{2^{n-1}}|_{n=1} = 2 - \frac{1}{2} - 1 = \frac{1}{2}$.
- *Cas inductiu:* $n > 1$. Tenim que $\sum_{i=1}^{n-1} \frac{i}{2^i} = 2 - \frac{n-1}{2^{n-1}} - \frac{1}{2^{n-2}}$. Per tant $\sum_{i=1}^{n} \frac{i}{2^i} = \frac{n}{2^n} + \sum_{i=1}^{n-1} \frac{i}{2^i} = \frac{n}{2^n} + 2 - \frac{n-1}{2^{n-1}} - \frac{1}{2^{n-2}} = \frac{n}{2^n} + 2 - \frac{2n-2}{2^n} - \frac{4}{2^n} = 2 + \frac{n-2n+2-4}{2^n} = 2 - \frac{n}{2^n} - \frac{1}{2^{n-1}}$.

(d) Si x és l'element $v[i]$, aleshores l'algorisme té cost $\Theta(i)$. Per tant el cost mig és

$$
\begin{aligned}
&\textstyle\sum_{0 \le i < n} \mathrm{Prob}(x = v[i]) \cdot \mathrm{Cost}(x = v[i]) = \\
&\textstyle\sum_{0 \le i < n} \mathrm{Prob}(x = v[i]) \cdot \Theta(i) = \\
&\textstyle\Theta(\sum_{0 \le i < n} \mathrm{Prob}(x = v[i]) \cdot i) = \\
&\textstyle\Theta(\sum_{1 \le i < n} \mathrm{Prob}(x = v[i]) \cdot i) = \\
&\textstyle\Theta(\mathrm{Prob}(x = v[n-1]) \cdot (n-1) + \sum_{1 \le i < n-1} \mathrm{Prob}(x = v[i]) \cdot i) = \\
&\textstyle\Theta(\frac{n-1}{2^{n-1}} + \sum_{1 \le i < n-1} \frac{i}{2^{i+1}}) = \\
&\textstyle\Theta(\frac{n-1}{2^{n-1}} + \frac{1}{2} \cdot \sum_{i=1}^{n-2} \frac{i}{2^i}) = \\
&\textstyle\Theta(\frac{n-1}{2^{n-1}} + \frac{1}{2} \cdot (2 - \frac{n-2}{2^{n-2}} - \frac{1}{2^{n-3}})) = \\
&\Theta(1)
\end{aligned}
$$

donat que $\lim_{n \to \infty} \frac{1}{2^n} = \lim_{n \to \infty} \frac{n}{2^n} = 0$.

Proposta de solució al problema 3

(a) Considerem cadascun dels casos:

- Assumim a, b parells. Tenim $\gcd(a/2, b/2) \mid a/2$, i així $2\gcd(a/2, b/2) \mid a$. Similarment, $2\gcd(a/2, b/2) \mid b$. Per tant $2\gcd(a/2, b/2) \mid \gcd(a, b)$. Per altra banda, com que a i b són parells, $gcd(a, b)$ és parell. Però $\gcd(a, b) \mid a$ implica que $\gcd(a, b)/2 \mid a/2$, i similarment $\gcd(a, b)/2 \mid b/2$. Per tant $\gcd(a, b)/2 \mid \gcd(a/2, b/2)$, i $\gcd(a, b) \mid 2\gcd(a/2, b/2)$, d'on finalment $\gcd(a, b) = 2\gcd(a/2, b/2)$.
- Assumim a senar i b parell. Per una banda $\gcd(a, b) \mid a$. Per una altra $\gcd(a, b) \mid b$, i com que a és senar, $\gcd(a, b) \mid b/2$. Així doncs tenim que $\gcd(a, b) \mid \gcd(a, b/2)$. I com que $\gcd(a, b/2) \mid a$ i $\gcd(a, b/2) \mid b$, tenim $\gcd(a, b/2) \mid \gcd(a, b)$. Per tant $\gcd(a, b) = \gcd(a, b/2)$.
- Assumim a, b senars i $a > b$. Si a i b són senars, $a - b$ és parell. Per tant $\gcd(a, b) = \gcd(a - b, b) = \gcd((a - b)/2, b)$ per la pista i l'apartat anterior.

(b) Una possible solució:

```
int gcd(int a, int b) {
  if (a == b)                        return a;
  if (a == 1 or b == 1)              return 1;
  if (a % 2 == 0 and b % 2 == 0)     return 2*gcd(a/2, b/2);
  if (a % 2 == 1 and b % 2 == 0)     return gcd(a,    b/2);
  if (a % 2 == 0 and b % 2 == 1)     return gcd(a/2, b    );
  else
    if (a > b) return gcd((a−b)/2, b    );
    else       return gcd(   a, (b−a)/2);
}
```

(c) Un cas pitjor es dóna, per exemple, quan a és una potència de 2 i b és senar. Aleshores a cada crida recursiva només decreix el primer argument, i b sempre és el segon argument. Quan el primer argument té i bits, el cost és $\Theta(i)$, per la divisió entre 2. Per tant el cost és $\sum_{i=1}^{n} \Theta(i) = \Theta(n^2)$.