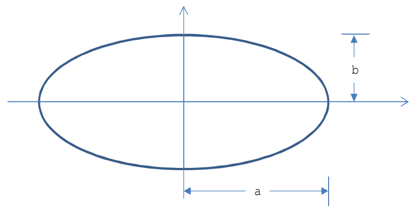
ISIS 1105 Diseño y Análisis de algoritmos
Semestre 2012-10 Parcial 2
Marzo 29, 2012
Prof. Rodrigo Cardoso

1 (30 puntos)

La figura representa una elipse, con ejes `a` y `b`, centrada en el origen de un plano cartesiano:



$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Los puntos `(x,y)` sobre la elipse satisfacen la ecuación indicada. Supóngase que `a`, `b` son enteros positivos y considérese el problema de determinar cuántos puntos `(i,j)`, con coordenadas enteras, están sobre la elipse. La siguiente especificación plantea una forma de resolver el problema (`S1`, `S2` y `S3` deberán desarrollarse después):

$[\mathrm{Ctx}\ C : 0<a \wedge 0<b$

$\quad\quad \mathbf{S1;}$

$\quad \{\mathrm{Inv}\ P:\ n + \#\{(i,j) \mid 0<p\le i<a \wedge 0<j\le q<b \wedge \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}$

$\quad\quad\quad\quad = \#\{(i,j) \mid 0<i<a \wedge 0<j<b \wedge \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}\}$

$\quad \{\mathrm{Cota}\ t:\ (a-p)*q\}$

$\quad\quad \mathbf{S2;}$

$\quad \{R1:\ n = \#\{(i,j) \mid 0<i<a \wedge 0<j<b \wedge \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}\ \}$

$\quad\quad \mathbf{S3}$

$\quad \{\mathrm{Pos}\ R:\ n = \#\{(i,j) \mid \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}\ \}$

$]$

1a (5/30) Explique el significado de `R1` y cómo se puede pasar fácilmente de `R1` a `R`.

`R1` resuelve el problema para el arco de elipse que está en el cuadrante en el que las coordenadas son positivas.

Nótese que los puntos `(a,0),(-a,0),(0,b),(0,-b)` están sobre la elipse. La respuesta que se encuentre en `R1`, multiplicada por `4` y aumentada en `4` es la respuesta que debe darse en `R`.

[5/5]

1b (5/30) Explique qué técnica de desarrollo de invariantes puede justificar el planteo de `P` con respecto a `R1`.

`P` es un balance de información.

[5/5]

- `n` es la respuesta parcial.
- $\#\{(i,j) \mid 0<i<a \wedge 0<j<b \wedge \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}\}$ es la respuesta en el cuadrante
- $\#\{(i,j) \mid 0<p\le i<a \wedge 0<j\le q<b \wedge \frac{i^2}{a^2} + \frac{j^2}{b^2} = 1\}$ es lo que no se ha calculado en el cuadrante

1c (5/30) Explique el significado de la cota t.

t mide el número de elementos en el área $\{(i,j) \mid 0<p\leq i<a \wedge 0<j\leq q<b\}$.

[5/5]

1d (10/30) Desarrolle código GCL para S1, S2 y S3.

```
[ Ctx C : 0<a ∧ 0<b

    S1: n,p,q:= 0,1,b-1;

   {Inv P: n + #{(i,j)| 0<p≤i<a ∧ 0<j≤q<b ∧ i²/a² + j²/b² = 1}
                        = #{(i,j)| 0<i<a ∧ 0<j<b ∧ i²/a² + j²/b² = 1}}
   {Cota t: (a-p)*q}

    S2: do p≠a ∧ q≠0 → if  a²p²+b²q² = a²b²  → n,p,q:= n+1,p+1,q-1
                       []  a²p²+b²q² > a²b²  → q:= q-1
                       []  a²p²+b²q² < a²b²  → p:= p-1
                       fi
        od

   {R1: n = #{(i,j)| 0<i<a ∧ 0<j<b ∧ i²/a² + j²/b² = 1} }

    S3: n:= 4*n + 4

   {Pos R: n = #{(i,j)| i²/a² + j²/b² = 1} }
]
```

[10/10]

1e (5/30) Considerando las asignaciones como operaciones básicas, calcule la complejidad temporal y espacial del algoritmo de 1d.

$T(a,b) = \theta(a+b)$
$\quad = \theta(\max(a,b))$

[3/5]

$S(a,b) = \theta(1)$.

[2/5]

Bonos

Adicionales por considerar mejoras para facilitar los cálculos.

(a) Asignaciones previas al ciclo, para evitar repeticiones de cálculos de constantes
En S1, calcular a^2, b^2, $a^2 * b^2$.

[+3]

(b) Fortalecimiento para calcular los valores de p^2, q^2 y $a^2i^2+b^2j^2$ a partir de valores anteriores. Conlleva asignaciones adicionales en las ramas de l condicional del cuerpo del ciclo e inicialización en S1.

[+3]

(c) Variable auxiliar para evitar repetición de código que calcula $a^2i^2+b^2j^2$. Conlleva uso de variable en el cuerpo del ciclo de S2.

[+1]

El siguiente programa implanta las mejoras indicadas:

```
[ Ctx C : 0<a ∧ 0<b

    S1: n,p,q:= 0,1,b-1;
        c,d:= a*a,b*b;
        e:= c*d;
        f,g:= 0,b*b-2*b+1

   {Inv P1: n + #{(i,j)| 0<p≤i<a ∧ 0<j≤q<b ∧ i²/a² + j²/b² = 1}
                      = #{(i,j)| 0<i<a ∧ 0<j<b ∧ i²/a² + j²/b² = 1}
              ∧ c=a² ∧ d=b² ∧ e=a²b²
              ∧ f=p² ∧ g=q² }
              }
   {Cota t: (a-p)*q}

    S2: do p≠a ∧ q≠0 →  h:= c*f+d*g
                        if  h = e
                            → n,p,q,f,g:=
                              n+1,p+1,q-1,f+2*p+1,g-2*q+1
                        []  h > e
                            → q,g:= q-1,g-2*q+1
                        []  h < e
                            → p,f:= p-1,f+2*p+1
                        fi
        od

   {R1: n = #{(i,j)| 0<i<a ∧ 0<j<b ∧ i²/a² + j²/b² = 1} }

    S3: n:= 4*n + 4

   {Pos R: n = #{(i,j)| i²/a² + j²/b² = 1} }
]
```

Castigo

Desarrollar un algoritmo correcto, pero no lineal

[-6]

2 (30 puntos)

Sea `nat⁺ ={1,2,3,...}`. Sean `n:nat⁺, b:[0..n-1] of nat⁺`.
Para `0≤i,j<n`, defínase `sumab(i,j) = (+k| i≤k≤j : b[k])`.

Escriba un algoritmo que solucione el siguiente problema:

```
[Ctx C: n: nat⁺ ∧ b:[0..n-1] of nat⁺
 {Pos R: r = #{(j,k): (j,k)∈0..n-1×0..n-1 ∧ sumab(0,j)=sumab(k,n-1)}
]
```

Incluya aserciones, invariantes y cotas que documenten su solución.

```
[Ctx C: n:nat⁺ ∧ b:[0..n-1] of nat⁺
```

```
p,q,r,sp,sq:= 0,n-1,0,b[0],b[n-1];
```

[2/30]

```
 {Inv P:  0≤p≤n ∧ -1≤q<n ∧
          r = #{(j,k): (j,k)∈0..p-1 × q+1..n-1 ∧ sumab(0,j)=sumab(k,n-1)}
          ∧ sp = sumab(0,p)
          ∧ sq = sumab(q,n-1)      }
```

[10/30]

```
{Cota t: n-p+q+1}
```

[3/30]

```
do
   p≠n ∨ q≠-1  →
```

[5/30]

```
   if    sp=sq →     r,p,q,sp,sq:= r+1,p+1,q-1,sp+b[p+1],sq+b[q-1]
   []    sp<sq →     p,sp:= p+1,sp+b[p+1]
   []    sp>sq →     q,sq:= q-1,sq+b[q-1]
   fi
```

[10/30]

```
od

 {Pos R: r = #{(j,k): (j,k)∈0..n-1 × 0..n-1 ∧ sumab(0,j)=sumab(k,n-1)}
]
```

Castigo

Desarrollar un algoritmo correcto, pero no lineal

[-6]

3 **(40 puntos)**

Una colonia de hormigas se reproduce por etapas y camadas. Sea `h(n,k)` la población en la camada `k` de la etapa `n`, para `n,k∈nat`.

La reproducción se da de acuerdo con las siguientes leyes:

(L1) La camada `0` de una etapa `n≥0`, tiene población `1`.

(L2) La camada `k` de una etapa `n`, con `0<k≤n`, tiene una población igual a la suma de las poblaciones en las camadas `k-1` y `k` de la etapa `n-1`.

(L3) La camada `k` de la etapa `n` tiene población `0`, si `0≤n<k`.

Se quiere determinar la población de la camada `s` en la etapa `r`, para ciertos `r,s` tales que `0≤s≤r`.

3a (30/40) Diseñe un algoritmo de programación dinámica para resolver el problema planteado.

3a1: Lenguaje

`h(n,k)` ≈ "población en la camada `k` de la etapa `n`", para `n,k∈nat`.

Se debe determinar `h(r,s)`.

[2/30]

3a2: Recurrencia

```
h(n,k)     = 1                        , si 0=k≤n
           = h(n-1,k-1) + h(n-1,k)  , si 0<k≤n
```

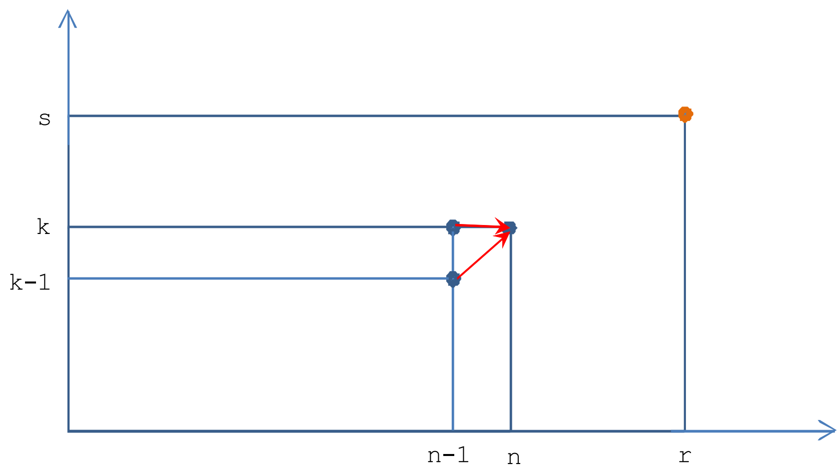
$= 0 \qquad\qquad\qquad , \text{ si } 0 \le n < k.$

[10/30]

N.B. Si se mira con atención, la recurrencia indicada corresponde con la que cumplen los coeficientes binomiales y que permiten definir el triángulo de Pascal, i.e.,

$$h(n,k) = \binom{n}{k}, \text{ para } n,k \in nat.$$

3a3: Diagrama de necesidades



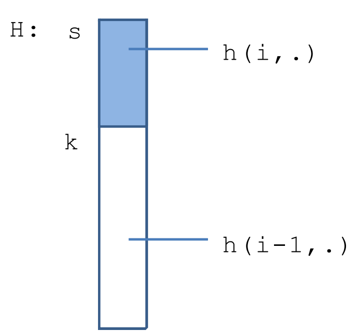
[8/30]

3a4: Estructura de datos + Invariante

El diagrama de necesidades sugiere una estructura de datos similar a las del problema del morral. Basta un vector columna `H[0..s]`, que almacene los valores de la función `h`, que se llena de arriba abajo:

[4/30]

`Inv:`



$\wedge \quad 0 < i \le r$

[6/30]

3b (10/40) Estime las complejidades espacial y temporal de su solución. Operación básica: asignación.

$T(r,s) = \theta(sr)$

Debe recorrerse la matriz del dominio de `h` ($0..r \times 0..s$), calculando $h(n,k)$ en $\theta(1)$.

Es fácil mejorar este recorrido ahorrándose $s^2/2$ cálculos, si se observa que $h(n,n)=1$. Esto permite inicializar el cálculo de las columnas `i`, $0 < i \le s$, cumpliendo `Inv`, así:

- en la columna 0: $h(0,0)=1$, $h(k,0)=0$, para $0<k\leq s$ (antes del ciclo)
- en la columna i: $h(i,i)=1$, $0<i\leq s$.

La mejora anterior, sin embargo, no es significativa, ya que también: $sr-s^2/2 = \theta(sr)$.

[5/10]

$S(r,s) = \theta(s)$

Tamaño del vector `H`.

[5/10]

N.B. Si se recuerda que $h(n,k)$ es un coeficiente binomial, es posible efectuar el cálculo de $h(r,s)$ de manera más eficiente, en tiempo $\theta(s)$. Sin embargo, esto se puede hacer usando propiedades específicas de los binomiales, no un algoritmo de programación dinámica.