# Pauta de Corrección

# Certamen Recuperativo
# Algoritmos y Complejidad

28 de diciembre de 2018

1. Las tres partes son independientes.

   *a*) Habiendo 3 coeficientes, esperamos una fórmula exacta hasta grado 2.

   *b*) Para hallar los coeficientes, podemos plantear un sistema de ecuaciones. Sabemos que:

   $$\begin{aligned} \int_{-} 1^1 x^k \,\mathrm{d}x &= \left. \frac{x^{k+1}}{k+1} \right|_{-1}^{1} \\ &= \frac{2}{k+1} [2 \mid k+1] \end{aligned}$$

   por lo que obtenemos:

   $$\begin{array}{rrrl} A_- & +A_0 & +A_+ & =2 \\ -A_- x_0 & & +A_+ x_0 & =0 \\ A_- x_0^2 & & +A_+ x_0^2 & =\frac{2}{3} \\ -A_- x_0^3 & & +A_+ x_0^3 & =0 \end{array}$$

   De la ecuación para $k = 1$ tenemos que $A_- = A_+$, con lo que inesperadamente se cumple para $k = 3$ también. En realidad, cumple para todo $k$ impar. El método es exacto hasta grado 3.
   Con la ecuación para $k = 0$ ahora tenemos que $A_0 = 2 - 2A_+$, la ecuación para $k = 2$ indica:

   $$A_+ = \frac{1}{3x_0}$$

   En resumen:

   $$\begin{aligned} A_- &= \frac{1}{3x_0} \\ A_0 &= 2 - \frac{2}{3x_0} \\ A_+ &= \frac{1}{3x_0} \end{aligned}$$

   Es claro que no necesariamente cumple para $k = 4$.

*c*) Si podemos elegir $x_0$ (idealmente $x_-$ y $x_+$, los puntos no necesariamente son simétricos en general), tenemos 5 parámetros, podemos esperar una fórmula exacta hasta grado 4. Por la observación anterior, será exacta hasta grado 5. Tenemos dos caminos:

- Usar la teoría de cuadratura gaussiana, desarrollando los polinomios ortogonales del caso y obteniendo los nodos como sus ceros.
- Extender el sistema de ecuaciones del punto anterior.

En nuestro caso es más fácil extender el sistema con la ecuación adicional:

$$A_- x_0^4 + A_+ x_0^4 = \frac{2}{5}$$

Conocemos $A_-, A_+$:

$$\frac{1}{3x_0} x_0^4 + \frac{1}{3x_0} x_0^4 = \frac{2}{5}$$

Directamente obtenemos:

$$x_0 = \sqrt[3]{\frac{3}{5}}$$

y resultan:

$$A_- = A_+ = \frac{\sqrt[3]{45}}{9}$$
$$A_0 = \frac{18 - 2\sqrt[3]{45}}{9}$$

## Puntajes

| | | |
|---|---|---|
| **Total** | | 25 |
| a) Grado máximo | 5 | |
| b) Obtención de coeficientes | 10 | |
| c) Mejor valor de $x_0$ | 10 | |

2. Vemos que interesa el largo del intervalo $[i, j-1]$ en el tiempo de ejecución. Sea $T(k)$ el tiempo de ejecución de SlowHeap$(i, j)$ cuando $j - i = k$. Del algoritmo, la recurrencia exacta es:

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil - 1) + \Theta(n)$$

Podemos aplicar el teorema maestro a la recurrencia aproximada:

$$T(n) = 2T(n/2) + \Theta(n)$$

Esto es $a = 2, b = 2, f(n) = \Theta(n)$, el valor crítico es $\alpha = \log_2 2 = 1$. Se aplica el cuarto caso del teorema maestro, con $\beta = 0$:

$$\begin{aligned} T(n) &= \Theta(n^\alpha \log^{\beta+1} n) \\ &= \Theta(n \log n) \end{aligned}$$

# Puntajes

| | | |
|---|---|---|
| **Total** | | 30 |
| Plantear la recursión | 10 | |
| Aplicar el teorema maestro | 20 | |

3. Corresponde hacer un análisis amortizado. Como hay una única operación, lo más sencillo es usar el método agregado. Sea $n$ el número máximo de widgets fabricados. Es claro que George Akeley hará un total de $n$ viajes, el costo por este concepto es $nv$. El número total de cambios de dígito es (el último dígito cambia $n$ veces, el dígito de las decenas $\lfloor n/10 \rfloor$ veces, y así sucesivamente):

$$\begin{aligned} n + \lfloor n/10 \rfloor + \lfloor n/10^2 \rfloor + \cdots &= \sum_{k \geq 0} \lfloor n/10^k \rfloor \\ &< \sum_{k \geq 0} n/10^k \\ &= n \sum_{k \geq 0} 10^{-k} \\ &= n \cdot \frac{1}{1 - 10^{-1}} \\ &= \frac{10n}{9} \end{aligned}$$

El costo total de $n$ operaciones está acotado por:

$$nv + \frac{10nc}{9} = n\left(v + \frac{10c}{9}\right)$$

con lo que el costo amortizado por operación está acotado por $v + 10c/9$.

## Puntajes

| | | |
|---|---|---|
| **Total** | | 30 |
| Usar método agregado | 5 | |
| Costo de viajes | 5 | |
| Costo de cambios de dígito | 10 | |
| Cotas manejables | 5 | |
| Costo amortizado | 5 | |

4. Por turno.

   *a*) El tiempo de ejecución del algoritmo no varía mayormente con la instancia presentada, y la tasa de error es menor a $1/2$. es un algoritmo de Monte Carlo.

   Su error es unilateral, con preferencia verdadero (*true biased*).

   *b*) En una corrida, la probabilidad de resultado correcto es $1/n$, por lo que la probabilidad de falla es $1-1/n$; en $k$ corridas independientes la probabilidad de error en todas es:

$$\left(1-\frac{1}{n}\right)^k$$

   Esto se parece seductivamente a la cota y límite:

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n=\frac{1}{\mathrm{e}}$$

   por lo que para $n$ grande la probabilidad de respuesta correcta en $n$ corridas independientes es:

$$1-\frac{1}{\mathrm{e}}$$

## Puntajes

| | | |
|---|---|---|
| **Total** | | 30 |
| a) Clasificar | 10 | |
| b) Análisis | 20 | |

HvB/LaTeX $2_\varepsilon$