



ALGORITMOS Y COMPLEJIDAD

Actividad 4 23 de abril de 2019

- **Objetivo:** Analizar eficiencia de un algoritmo, justificando de forma **completa y detallada** el orden exacto del tiempo y espacio de ejecución. Reconocer la estrategia de algoritmos greedy.
- **Metodología:**
 - Considere el siguiente algoritmo:

Algorithm 1 function $f(x:\text{integer})$

```
 $n \leftarrow \lceil \log_2 x \rceil$ 
 $d[1..n+1]$ , donde  $d[i] \leftarrow 2^{i-1}$ ,  $1 \leq i \leq n+1$ 
 $a[1..n+1]$ , donde  $a[i] \leftarrow 0$ ,  $1 \leq i \leq n+1$ 
 $i \leftarrow n+1$ 
while  $i \geq 1$  do
  while  $x \geq d[i]$  do
     $x \leftarrow x - d[i]$ 
     $a[i]++$ 
  end while
   $i--$ 
end while
return  $a$ 
```

donde por ejemplo $f(95)$ devuelve $a = [1, 1, 1, 1, 1, 0, 1, 0]$, y como $n = \lceil \log_2 95 \rceil = 7$ entonces tanto d como a tienen dimensión 8.

- Responda las siguientes preguntas, justificando completamente su respuesta:
 1. Identificar el problema resuelto y la estrategia Greedy utilizada.
 2. Identificar las porciones del código donde implícitamente se define el conjunto de candidatos y las funciones Greedy: `esViable()`, `esSolucion()` y `selección()`.
 3. Analizar el tiempo de ejecución del algoritmo, dando las cotas más precisas. El resultado deberá ser expresado en función de la cantidad de dígitos de la entrada, asumiendo x representado en notación decimal. Las operaciones aritméticas se consideran de tiempo constante.
 4. Similarmente, analizar el tiempo de ejecución del algoritmo si para cualquier entrada x , asumimos sólo vectores a y d con dimensión 1, donde $d[1] \leftarrow 1$. Ejemplo: $f(95)$ retorna $a[1] = 95$.
- **Evaluación:** La aprobación de esta actividad otorgará hasta 1 crédito en CT3 (performance en tiempo y espacio de algoritmos).
- **Observación:** Se recuerda que la actividad es optativa.