Análisis de Algoritmos 2024

Trabajo Práctico 2 - Notación Asintótica 1

Fecha de entrega: 30 de Agosto de 2024

El objetivo de este trabajo es realizar los ejercicios propuestos de forma colaborativa. Para los ejercicios compartidos entre grupos, pueden compartir el desarrollo del ejercicio o realizar dos resoluciones diferentes.
En particular para este práctico se pide en algunos casos tener código funcionando, verifica tener una computadora que ejecute tu código si te tocó algún ejercicio de este tipo. Se pedirá ejecución en clase.
Todos los ejercicios deben ser resueltos por todos los estudiantes, pero en este trabajo sólo deben ser publicados los que te tocaron según la asignación de la tabla siguiente. Dicha asignación corresponde a la presentación oral del TP. Si el día de la práctica hay varias resoluciones logradas, elegiremos 1 grupo para exponer ese ejercicio.
Por supuesto, si querés compartir la resolución de otros ejercicios en este proyecto Overleaf, también son bienvenidas pero no es obligatorio.

Nota: Colocá el nombre del/de los estudiante/s que resolvió/resolvieron el ejercicio al iniciar la resolución del ejercicio que te tocó.

Las respuestas deben quedar escritas en este mismo .tex, en Overleaf.

Grupos (Estudiantes por grupo)	Ej. asignado
Grupo 1 (Sthefany, Victoria, Paula, Adriano, Facundo)	1.1 || 2.1 || 2.2|| 3.1 || 3.3.h || 3.4.a
Grupo 2 (Sebastián, Bautista, Antonio, Albany, Luis)	1.2 || 2.3 || 3.2.a || 3.3.a || 3.3.g || 3.4.b
Grupo 3 (Lucas, Valentina, Nicanor, Ignacio, Franco)	1.3|| 2.4 || 3.2.b || 3.3.f || 3.4.c
Grupo 4 (Manuel, Jan, Ulises, Demian , Luciano)	2.5 || 3.2.c || 3.3.e || 3.4.d || 4.3|| 4.1.iii
Grupo 6 (Roman, Ulises, Guillermo, Jamiro, Juan)	2.4 || 3.2.d || 3.3.d || 3.4.a || 4.2
Grupo 7 (Christopher, Tomás, Joaquín, Leonard, Facundo)	2.3 || 3.2.e || 3.3.c || 3.4.b || 4.1.i
Grupo 8 (Facundo, Belén, Jeremías, Bruno, Kevin)	2.2 || 3.2.d || 3.3.b || 3.4.c || 4.1.ii

1. Preguntas teóricas

1. Dado un conjunto de algoritmos que resuelven el mismo problema, ¿qué consideraciones usaría para elegir uno?.
2. ¿Qué es una prueba empírica? ¿Para qué sirve? Indique un ejemplo de un código no correcto, otro no eficiente que haya dado buenos resultados empíricos.
3. Si analiza un algoritmo de acuerdo a la periodicidad o frecuencia de uso, ¿qué puede decir?

2. Algoritmos y su notación asintótica

1. Especifique un código que a partir de un valor de entrada n tenga un orden constante.

   a) Indique tiempos de ejecución.

   b) Ejecute dicho algoritmo con diferentes valores de entrada

2. Especifique un código que realice algo específico que tenga un orden $O(n)$.

   a)Ejecútelo con un n=10000.

3. Realice lo mismo para un algoritmo de $O(n^2)$

   a)Indique tiempo empírico con n = 10000.

4. Ejecute un algoritmo de $O(logn)$

   a)Indique tiempo empírico con n = 10000.

5. Especifique un algoritmo conocido (o no) que realice tenga parte de $O(1)$, parte de $O(n)$ y parte de $O(nlogn)$. Cuál es el orden de todo el algoritmo?

   a)Ejecútelo con un n=10000.

3. Notación Asintótica

1. Supongamos tener dos algoritmos para el mismo problema: el algoritmo A insume tiempo n, mientras que el algoritmo B toma tiempo $3\ log\ n + 5$. ¿Cuándo se debería elegir A y cuándo B?

2. Demostrar las siguientes propiedades:

   a) (reflexividad) $f(n) \in O(f(n))$

   b) (transitividad) $f(n) \in O(g(n))\ y\ g(n) \in O(h(n)) \Rightarrow f(n) \in O(h(n))$

   c) $O(f(n)) = O(g(n)) \Leftrightarrow f(n) \in O(g(n))$ y $g(n) \in O(f(n))$

   d) $f(n) \in O(h(n))\ y\ g(n) \in O(h(n)) \Rightarrow (f+g)(n)) \in O(h(n))$

   e) $f(n) \in O(h(n))\ y\ g(n) \in O(l(n)) \Rightarrow (f*g)(n)) \in O(h(n)*l(n))$

3. Verificar que:

   (a) $1000000 \in O(1)$

   (b) $10n^2 \in O(n^3)$

   (c) $2^n \in O(n^n)$

   (d) $n! \in O(n^n)$

   (e) $\sum_{k=0}^{n} k \in O(n^2)$

   (f) $\sum_{k=0}^{n} k^2 \in O(n^3)$

   (g) $\sum_{k=r} k^3 \in O(n^4)$

   (h) (...)= $\frac{m}{k} = 0^m a_k n^k$ $(a_k \in \mathbb{R})$, entonces $(n) \in O(n^m)$

4. Analizar si cada una de las siguientes afirmaciones es verdadera o falsa, argumentando apropiadamente las respuestas dadas:
   (a) $2^{n+1} \in O(2^n)$
   (b) $2^{2n} \in O(2^n)$
   (c) $O(1) \subseteq O(2^n)$
   (d) Si $f(n) \in O(g(n))$, entonces $log\ f(n) \in O(log\ g(n))$

4. Algoritmia en Algoritmos iterativos sencillos

1. Para cada una de las siguientes funciones:
   (a) averiguar el tiempo teórico de cada una de ellas. Expresar la respuesta como función de n
   (b) ¿qué es lo que calculan? Expresar la respuesta en función de n.
   (i)

```
int f (int n){
    int sum=0;
    for(int i=1;i<=n;++i)
        sum+=i;
    return sum;
}
```

   (ii)

```
int g(int n){
    int sum=0;
    for(int i=1;i<n;++i)
        sum+=i+f(n);
    return sum;
}
```

   (iii)

```
int h(int n){
    return f(n)+g(n);
}
```

2. Demostrar formalmente si existe relación de pertenencia entre $f(n)$ y $O(g(n))$ y también entre $g(n)$ y $O(f(n))$ considerando $f(n) = T(n)$ y $g(n) = n^3$, donde $T(n)$ es la función resultante del ejercicio *2.1.a.iii* anterior.

3. Demostrar formalmente si existe relación de pertenencia entre $f(n)$ y $O(g(n))$ y también entre $g(n)$ y $O(f(n))$ considerando $f(n) = T(n)$ y $g(n) = log\ n$, donde $T(n)$ es la función resultante del ejercicio *2.1.a.i* anterior.