

Dependencia	Facultad de Ingeniería	Programa	Ingeniería en Software
Asignatura	Análisis de Algoritmos	Semestre	7
Jornada	Diurna	Grupo	PREPRO25020020015
Docente	Jaime E. Montoya M.	Fecha entrega	02/09/2025
Actividad	Tarea 1	Unidad	1
Valor	20%	Calificación	
Estudiante		Documento de identidad	

Notas

Lea atentamente las indicaciones que se indican a continuación, así como cada enunciado de los problemas planteados:

- En la calificación se tiene en cuenta las buenas prácticas:
 - Reglas de estilos: orden en el código, uso de notación para nombres de variables (*camelCase*, *snake_case* notación húngara, etc.), indentación del código, nombres adecuados para archivos y algoritmos siguiendo reglas para nombres de campos, etc.
 - Puntualidad en la entrega
 - Eficiencia del desarrollo
 - Realizar lo que se pide y no utilizar técnicas más avanzadas a las vistas. Se deben hacer los desarrollos algorítmicos y no funciones que pueda poseer el lenguaje para solucionar algún requerimiento solicitado
- La actividad se desarrolla en forma individual o en parejas
- Algunos casos pueden ser presentados algorítmicamente
- Cada respuesta debe ser justificada, ya sea matemática o algorítmicamente
- La actividad está basada en el proyecto realizado en clase llamado que se encuentra en GitHub (tanto para Python como para Java), por lo que debe tener en cuenta esto para la solución de la actividad haciendo uso de los métodos y atributos a que haya lugar, además de los que requiera crear para solucionar los puntos planteados. Se sugiere crear opciones de menú para los puntos de la actividad a dónde dé lugar. También tiene la opción de desarrollar solo las funciones que se requieren para dar solución a la actividad
- La entrega se realiza enviando a la plataforma Virtual todos los desarrollos en una carpeta comprimida en un solo archivo en formato *zip*. Puede también enviar un documento sólo con los métodos requeridos basados en las clases en mención
- Los problemas deben ser desarrollados usando el lenguaje Java y Python, según sea el caso o de acuerdo con el problema planteado
- Los ejercicios pueden ser revisados en clase para ser sustentados por los estudiantes
- El archivo zip enviado a la plataforma deben nombrarlo con el nombre y apellido del estudiante siguiendo las normas para los nombres de campos. Por ejemplo: **PepitoPerez_AnaGil.zip**

10. Se tiene en cuenta la creatividad a la hora de desarrollar los problemas, pensando en la *usabilidad*, esto es, en programas fáciles de usar para el usuario en cuanto a lo operativo, intuitivo, presentación y otros aspectos

Actividad

- 1.(Valor 1.0) Suponga que varios algoritmos se ejecutan y analizan, entregando una función de tiempo para cada uno indicadas a continuación. Demuestre en cada caso que para el tiempo de ejecución $f(n)$ dado por el respectivo contador de frecuencias de cada algoritmo:
 - a. $f(n) = 2n + 4$ siempre es un número es un par independientemente de que n sea par o impar
 - b. $f(n) = 4n + 5$ siempre es un número es un impar independientemente de que n sea par o impar
 - c. $f(n) = 6n^2 - 2n + 1$ siempre es un número es un impar independientemente de que n sea par o impar
2. (Valor 1.0) Demuestre usando la inducción matemática que, si n es un número entero, entonces $2n + 2$ es un número par
3. (Valor 1.0) Analice el ejemplo 0.10 de las notas de clase e indique:
 - a. Qué tarea realiza el algoritmo y cómo la hace (especifique que hace cada operación)
 - b. Contador de frecuencias y orden de magnitud del algoritmo
- 4.(Valor 1.0) Dado un valor real entre 0 y 2π , cree dos funciones para calcular el **seno y coseno** trigonométricos en radianes usando la serie de Taylor para n términos, donde n es una entrada grande:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$
 - a. Determine el contador de frecuencias de estas funciones y su orden de magnitud
 - b. Determine el porcentaje de error entre el cálculo hecho con la serie de Taylor y los valores entregados por los lenguajes de programación (observe cómo la precisión aumenta a medida que $n \rightarrow \infty$)
- 5.(Valor 1.0) Sean $f(x)$ y $g(x)$ dos funciones polinómicas cualesquiera. Cree un programa para sumar y restar dos **polinomios** utilizando listas ligadas. Si a es una constante, entonces calcule $f(a)$, $g(a)$, además, derive a $f(x)$ y a $g(x)$ y encuentre $f'(a)$ y $g'(a)$.
 - a. Determine el contador de frecuencias de este algoritmo y su orden de magnitud
 - b. Determine cuánta memoria este algoritmo
 - c. Determine el porcentaje de error entre el cálculo hecho con la serie de Taylor y los valores entregados por los lenguajes de programación (observe cómo la precisión aumenta a medida que $n \rightarrow \infty$)
- 6.(Valor 1.0) Encuentre los **números perfectos** entre 1 y n cuando $n \rightarrow \infty$. Recuerde que un número natural es perfecto si sus divisores excepto el mismo número, suman dicho

número, por ejemplo, los divisores de 6 son 1, 2, 3 y 6, descartamos el 6 y encontramos que $1 + 2 + 3 = 6$, por lo que dicho número es perfecto

- a. Determine el contador de frecuencias de este algoritmo y su orden de magnitud
 - b. Mida los tiempos de ejecución que entrega el sistema operativo en Java y Python para ejecutar este programa, respectivamente
 - c. ¿Cuáles números y cuántos cumplen esta característica de los números naturales y en qué intervalo?
 - d. ¿Cuál es la memoria utilizada en este programa?
- 7.(Valor 1.0) Un centro de investigación guarda el registro de las temperaturas tomadas constantemente en una ciudad durante el año y han realizado el registro por varios años conservando el histórico. Cree un programa para simular la situación generando temperaturas aleatoriamente (entre -10° y 45°) para 3 años suponiendo que en éstos se han registrado grandes cantidades de datos (esto es, valores de n grandes: $\geq 10^6$). Realice lo siguiente:
- a. Promedie las temperaturas por año. “Juegue” con los valores de entrada entre pequeños, medianos y grandes (para los tres años) y mida los tiempos de ejecución que entrega el sistema operativo para cada parte de interés del programa indicando las conclusiones sobre los resultados ¿el cálculo del promedio es una operación elemental? Indique cuáles datos de entrada utilizó y qué resultados obtuvo de forma tabular
 - b. Encuentre el contador de frecuencias y el orden de magnitud O del programa
- 8.(Valor 1.0) Para entradas grandes de n (tales como 10^3 , 10^4 , 10^5 , etc.), muestre gráficamente que $O(1) < O(\lg(n)) < O(n) < O(n^2) < O(2^n)$ (Sugerencia: el papel milimetrado facilita la creación de las gráficas)
- 9.(Valor 1.0) Uno de los algoritmos más populares para ordenar vectores (listas) es conocido con el nombre de “Burbuja”, y más formalmente, como “método de intercambio directo”, se presenta más abajo. Es un algoritmo fácil de implementar, pero también es el más ineficiente de todos los métodos de ordenación conocidos. Se pide a partir de él:
- a. Encontrar el contador de frecuencias
 - b. Determinar el orden de magnitud O (Big O)
 - c. Indicar cuál es la memoria utilizada
 - d. Utilizar las funciones del sistema operativo suministradas por el lenguaje para medir el tiempo de ejecución de este algoritmo para entradas grandes de n , tales como 10^4 , 10^5 , 10^6 , etc., tabule y grafique los datos encontrados y compare el rendimiento entre Java y Python, probando en máquinas con características de hardware distintas y preferiblemente en distintos sistemas operativos

Algoritmo 1:

```
public void sortVector()
{
    int aux;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (vec[i] > vec[j]) {
```

```

        aux = vec[i];
        vec[i] = vec[j];
        vec[j] = aux;
    }
}
}
}

```

10. (Valor 1.0) Para los siguientes algoritmos:

- Encontrar el contador de frecuencias
- Determinar el orden de magnitud O (Big O)
- Indicar cuál es la memoria utilizada
- Graficar las funciones del tiempo de ejecución de cada algoritmo

Algoritmo 2

```

Inicio
Enteros: c, i, j, n, num
Leer n
i = 1
Mientras i <= n
    j = i
    c = 0
    Mientras j > 1 Hacer
        leer num
        c = c + num * j ^ 2
        j = j / 2
    FinMientras
    Imprimir "Valor de c: ", c
FinMientras
Fin

```

Algoritmo 3

```

Inicio
Enteros: hh, mm, ss, h, m, s
Leer hh, mm, ss
h = 0
Mientras h < hh
    m = 0
    Mientras m < mm Hacer
        s = 0
        Mientras s < ss Hacer
            Imprimir h + ":" + m + ":" + s + "\n"
            s = s + 1
        FinMientras
        m = m + 1
    FinMientras
    h = h + 1

```

FinMientras
Fin