Comenzado el	lunes, 8 de octubre de 2018, 14:15
Estado	Finalizado
Finalizado en	lunes, 8 de octubre de 2018, 15:25
Tiempo empleado	1 hora 9 minutos
Puntos	14/14
Calificación	**10** de 10 (**100**%)

Pregunta 1

Correcta

Puntúa 1 sobre 1

Se tiene un algoritmo que realiza cierta cantidad de procesos sobre un conjunto de ***n*** datos y un minucioso análisis matemático ha determinado que la cantidad de procesos que el algoritmo realiza en el peor caso viene descripto por la función $f(n) = 3n^3 + 5n^2 + 2n^{1.5}$ ¿Cuál de las siguientes expresiones representa mejor el orden del algoritmo para el peor caso?

Seleccione una:

a. $O(3n^3 + 5n^2 + 2n^{1.5})$

b. $O(n^{1.5})$

c. $O(n^3 + n^2)$

d. $O(n^3)$ ✓

¡Correcto!

¡Correcto!

La respuesta correcta es:
$O(n^3)$

Pregunta 2

Correcta

Puntúa 2 sobre 2

En la columna de la izquierda se muestra el código fuente en Python de diversos procesos sencillos. Para cada uno de ellos, seleccione de la lista de la derecha la expresión en notación *Big O* que mejor describa el tiempo de ejecución de cada proceso para el peor caso. (Aclaración: una expresión como *n^2* debe entenderse como "*n al cuadrado*" o n^2).

```
n = int(input('N: '))
ac = 0
for i in range(n):
    for j in range(i+1, n):
        ac += i*j
print(ac)
```

O(n^2) ▾ ✓

```
n = int(input('N: '))
ac = 0
for i in range(n):
    ac += i
print(ac)
```

O(n) ▾ ✓

```
n = int(input('N: '))
ac = 0
for i in range(n):
    for j in range(i+1, n):
        ac += i*j

for i in range(n):
    for j in range(n):
        for k in range(n):
            ac += (i+j+k)
print(ac)
```

O(n^3) ▾ ✓

```
ac = 0
for i in range(10):
    ac += i
print(ac)
```

O(1) ▾ ✓

¡Correcto!

La respuesta correcta es:

```
n = int(input('N: '))
ac = 0
for i in range(n):
    for j in range(i+1, n):
        ac += i*j
print(ac)
```

→ O(n^2),

```
n = int(input('N: '))
ac = 0
for i in range(n):
    ac += i
print(ac)
```

→ O(n),

```
n = int(input('N: '))
ac = 0
for i in range(n):
    for j in range(i+1, n):
        ac += i*j

for i in range(n):
    for j in range(n):
        for k in range(n):
            ac += (i+j+k)
print(ac)
```

→ O(n^3),

```
ac = 0
for i in range(10):
    ac += i
print(ac)
```

→ O(1)

Pregunta 3

Correcta

Puntúa 2 sobre 2

Analice el siguiente esquema de una función en Python:

```
def procesar(n, m):
    for i in range(n+1):
        for j in range(m+1):
            # ... acciones sencillas a realizar...
            # ... suponga que no hay otro ciclo aq
uí...
            # ... y que sólo aparecen operaciones
 de tiempo constante...
```

¿Cuál de las siguientes expresiones de orden describe mejor el tiempo de ejecución de esta función en el peor caso?

Seleccione una:

a.
`O(n*n)`

b.
`O(n*m)` ✓

¡Correcto! Efectivamente, no hay problema alguno en que una expresión de orden se base en dos o más variables si el tamaño del problema depende de esas variables.

c.
`O(n)`

d.
`O(m)`

¡Correcto!

La respuesta correcta es:
`O(n*m)`

Pregunta 4

Correcta

Puntúa 1 sobre 1

¿Cuál es la ***principal*** característica de todos los métodos de ordenamiento conocidos como métodos simples o directos?

Seleccione una:

a.
Son muy fáciles de programar.

b.
Son muy veloces para cualquier tamaño del arreglo a ordenar.

c.
Tienen un tiempo de ejecución de orden cuadrático. ✓

¡Correcto!

d.
Tienen un tiempo de ejecución de orden lineal.

¡Correcto!

La respuesta correcta es:
Tienen un tiempo de ejecución de orden cuadrático.

Pregunta 5

Correcta

Puntúa 1 sobre 1

¿Cuáles de las siguientes son características **correctas** del algoritmo *Shellsort*? (Más de una puede ser cierta... marque TODAS las que considere válidas)

Seleccione una o más de una:

a.

En el caso promedio, el algoritmo *Shellsort* es tan eficiente como el *Heapsort* o el *Quicksort*, con tiempo de ejecución *O(n*log(n))*.

b.

El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Selección Directa*, consistente en buscar iterativamente el menor (o el mayor) entre los elementos que quedan en el vector, para llevarlo a su posición correcta, pero de forma que la búsqueda del menor en cada vuelta se haga en tiempo logarítmico.

c.

El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Inserción Directa* (o *Inserción Simple*), consistente en armar suconjuntos ordenados con elementos a distancia $h > 1$ en las primeras fases, y terminar con $h = 1$ en la última. ✓

¡Correcto!

d.

El algoritmo *Shellsort* es complejo de analizar para determinar su rendimientos en forma matemática. Se sabe que para la serie de incrementos decrecientes usada en la implementación vista en las clases de la asignatura, tiene un tiempo de ejecución para el peor caso de $O(n^{1.5})$. ✓

¡Correcto!

¡Correcto!

Las respuestas correctas son:
El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Inserción Directa* (o *Inserción Simple*), consistente en armar suconjuntos ordenados con elementos a distancia $h > 1$ en las primeras fases, y terminar con $h = 1$ en la última.,

El algoritmo *Shellsort* es complejo de analizar para determinar su rendimientos en forma matemática. Se sabe que para la serie de incrementos decrecientes usada en la implementación vista en las clases de la asignatura, tiene un tiempo de ejecución para el peor caso de $O(n^{1.5})$.

Pregunta 6

Correcta

Puntúa 1 sobre 1

¿Qué diferencia existe entre el *conteo exhaustivo* de operaciones críticas y el *análisis asintótico* del comportamiento de una función en el análisis de algoritmos?

Seleccione una:

a.
Ninguna. Ambas se refieren a la misma técnica básica del análisis de algoritmos

b.
El *conteo exhaustivo* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones críticas que lleva a cabo un algoritmo, mientras que el *análisis asintótico* busca determinar el comportamiento general de una función para valores muy grandes del tamaño del problema. ✓

¡Correcto!

c.
El *análisis asintótico* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones críticas que lleva a cabo un algoritmo, mientras que el *conteo exhaustivo* busca determinar el comportamiento general de una función para valores muy grandes del tamaño del problema.

d.
El *conteo exhaustivo* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones ***críticas*** que lleva a cabo un algoritmo, mientras que el *análisis asintótico* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones ***no críticas*** que lleva a cabo un algoritmo.

¡Correcto!

Las respuestas correctas son:
El *conteo exhaustivo* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones críticas que lleva a cabo un algoritmo, mientras que el *análisis asintótico* busca determinar el comportamiento general de una función para valores muy grandes del tamaño del problema.,

El *conteo exhaustivo* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones ***críticas*** que lleva a cabo un algoritmo, mientras que el *análisis asintótico* busca determinar una expresión o fórmula que exprese de manera rigurosa la cantidad de operaciones ***no críticas*** que lleva a cabo un algoritmo.

Pregunta 7

Correcta

Puntúa 1 sobre 1

¿Qué se entiende, en el contexto del Análisis de Algoritmos, por un *Orden de Complejidad*?

Seleccione una:

a.
Un conjunto o familia de funciones matemáticas que se comportan asintóticamente de la misma forma. ✓

¡Correcto!

b.
Un conjunto o familia de algoritmos que resuelven el mismo problema.

c.
Un conjunto o familia de subrutinas con similares objetivos (equivalente al concepto de *módulo*).

d.
Un conjunto de datos ordenados.

¡Correcto!

Las respuestas correctas son:
Un conjunto o familia de funciones matemáticas que se comportan asintóticamente de la misma forma.,

Un conjunto o familia de algoritmos que resuelven el mismo problema.

Pregunta 8

Correcta

Puntúa 1 sobre 1

Si los algoritmos de *ordenamiento simples* tienen todos un tiempo de ejecución $O(n^2)$ en el peor caso, entonces: ¿cómo explica que las mediciones efectivas de los tiempos de ejecución de cada uno sean diferentes frente al mismo arreglo?

Seleccione una:

a.
La notación *Big O* rescata el término más significativo en la expresión que calcula el rendimiento, descartando constantes y otros términos que podrían no coincidir en los tres algoritmos. ✓

¡Correcto!

b.
Los tiempos deben coincidir. Si hay diferencias, se debe a errores en los intrumentos de medición o a un planteo incorrecto del proceso de medición.

c.
La notación *Big O* no se debe usar para estimar el comportamiento en el peor caso, sino sólo para el caso medio.

d.
La notación *Big O* no se usa para medir tiempos sino para contar comparaciones u otro elemento de interés. Es un error, entonces, decir que los tiempos tienen "*orden n cuadrado*".

¡Correcto!

La respuesta correcta es:
La notación *Big O* rescata el término más significativo en la expresión que calcula el rendimiento, descartando constantes y otros términos que podrían no coincidir en los tres algoritmos.

Pregunta 9

Correcta

Puntúa 1 sobre 1

¿Cuáles de las siguientes *son correctas* en cuanto a los tiempos de ejecución de los algoritmos de ordenamiento clásicos? (Más de una puede ser cierta... marque TODAS las que considere válidas)

Seleccione una o más de una:

- [x] a.
  Algoritmo Quick Sort: O(n*log(n)) en el caso promedio, pero $O(n^2)$ en el peor caso. ✓

  ¡Correcto!

- [x] b.
  Algoritmo Heap Sort: O(n*log(n)) tanto para el caso promedio como para el peor caso. ✓

  ¡Correcto!

- [ ] c.
  Algoritmo Shell Sort: $O(n^2)$ en el peor caso para la serie de incrementos decrecientes vista en clase.

- [x] d.
  Algoritmos directos o simples: $O(n^2)$ en el peor caso para todos ellos. ✓

  ¡Correcto!

¡Correcto!

Las respuestas correctas son:
Algoritmos directos o simples: $O(n^2)$ en el peor caso para todos ellos.,

Algoritmo Quick Sort: O(n*log(n)) en el caso promedio, pero $O(n^2)$ en el peor caso.,

Algoritmo Heap Sort: O(n*log(n)) tanto para el caso promedio como para el peor caso.

Pregunta 10

Correcta

Puntúa 3 sobre 3

Si se realiza un análisis preciso del ordenamiento por *Selección Directa* para un arreglo de *n* componentes, se llega a la conclusión que ese algoritmo hará *n-1* pasadas, con *n-1* comparaciones en la primera, *n-2* en la segunda, y así sucesivamente reduciendo de a 1 la cantidad de comparaciones hasta hacer sólo una comparación en la última pasada. Por lo tanto, el algoritmo hará *invariablemente* una cantidad total de $\frac{1}{2}(n^2 - n)$ comparaciones. Sabiendo esto, ¿cuáles de las siguientes expresiones son correctas para describir la cantidad de comparaciones que hará el algoritmo, usando distintos tipos de notaciones? (Más de una respuesta puede ser correcta. Marque TODAS las que considere correctas)

Seleccione una o más de una:

- [x] a.
  Cantidad de comparaciones: $\Theta(n^2)$ ✓
  ¡Correcto!
- [x] b.
  Cantidad de comparaciones: $\Omega(n^2)$ ✓
  ¡Correcto!
- [ ] c.
  Cantidad de comparaciones: $o(n^2)$
- [x] d.
  Cantidad de comparaciones: $O(n^2)$ ✓
  ¡Correcto!

¡Correcto!

Las respuestas correctas son:
Cantidad de comparaciones: $O(n^2)$,
Cantidad de comparaciones: $\Omega(n^2)$,
Cantidad de comparaciones: $\Theta(n^2)$