

Examen 1 - Requires Respondus LockDown Browser

- Fecha de entrega 24 de sep en 15:00
- Puntos 100
- Preguntas 20
- Disponible después de 24 de sep en 11:10
- Límite de tiempo 90 minutos
- Es obligatorio utilizar el navegador Respondus LockDown Browser

Historial de intentos

	Intento	Hora	Puntaje
MÁS RECIENTE	Intento 1	63 minutos	55.16 de 100

Puntaje para este examen: 55.16 de 100

Entregado el 24 de sep en 12:17

Este intento tuvo una duración de 63 minutos.



Pregunta 1

0 / 5 pts

¿Cual es el Z-array para encontrar "pay" en "payday"?

Respondido

☒ 000123000

☐ 0000123000

Respuesta correcta

☐ 0000300000

☐ 000300000



Pregunta 2

0 / 5 pts

Este es el ciclo principal del algoritmo de Manacher.

Considera que $T = @\$a\$b\$b\$a\$#$ es la cadena con caracteres extra sobre la cual se realiza el calculo.

El ciclo while (mientras) realiza las comparaciones.

¿Qué va en el espacio en blanco?

e = tamaño de T

para i = 1 hasta e-1:

si i < limite:

espejo = 2 * centro - i

p[i] = min(limite - i, p[espejo])

gap = (p[i] + 1)

mientras _____

p[i] += 1

gap += 1

Si i + p[i] > limite:

centro = i

limite = i + p[i]

☐ T[limite] == T[gap]

Respondido

☒ T[gap] == T[i + gap]

Respuesta correcta

☐ T[i - gap] == T[i + gap]

☐ i < limite



Pregunta 3

2.5 / 5 pts

¿Cual es el arreglo LPS de "onions"?

[0, 0, 0, 1, 2, 0]

[0, 0, 1, 2, 0, 0, 0, 1, 2, 1, 2]

Respuesta 1:

[0, 0, 0, 2, 0, 0]

[0, 0, 0, 2, 2, 0]

[0, 0, 0, 0, 2, 0]

¡Correcto!

[0, 0, 0, 1, 2, 0]

Respuesta 2:

[0, 0, 2, 1, 0, 0, 0, 4, 3, 2, 1]

Respondido

[0, 0, 1, 2, 0, 0, 0, 1, 2, 1, 2]

Respuesta correcta

[0, 0, 1, 2, 0, 0, 0, 1, 2, 3, 4]

[1, 2, 1, 2, 0, 0, 0, 1, 2, 3, 4]



Pregunta 4

5 / 5 pts

Ayuda a cada persona con el algoritmo mas adecuado de entre las siguientes opciones.

A) Luis quiere conocer las rutas mas cortas para que 5 amigos lleguen desde sus casas a casa de Luis.

B) Karla debe conectar 5 sub-estaciones, de modo que formen una red conexas sin ciclos, y usando la menor cantidad de cable.

C) Lola debe crear un plan de entregables de proyecto. Algunos entregables dependen de otros, que deben estar listos antes.

¡Correcto!

Luis

Algoritmo de Dijkstra, cuyc



¡Correcto!

Karla

Algoritmo de Prim, cuyo er



¡Correcto!

Lola

Algoritmo de ordenamientc



Otras opciones de coincidencia incorrecta:

- Algoritmo de TSP, cuyo enfoque es greedy
- Algoritmo KMP, cuyo enfoque es programación dinámica
- Algoritmo de detección de ciclos, cuyo enfoque es divide and conquer



Sin responderPregunta 5

0 / 5 pts

Ana esta organizando una carne asada en su casa, y quiere que el mayor número de gente asista.

El problema es que algunos potenciales invitados odian a otros, y no asistirán si alguien a quien odian fue invitado también.

Cada invitado tiene una lista de personas que le odian, por ejemplo:

Luis es odiado por Fer y Nico

Nico es odiado por Fer

Mario es odiado por Lola, Nico, Bob, Sam y Carlos

Carlos es odiado por Max

Fer es odiado por Nico

etc, etc...

Diseña el **pseudocódigo** de un algoritmo greedy para el problema de Ana, incluyendo la descripción de la representación de las soluciones. Asume que a Ana no la odia nadie.

Su respuesta:



Pregunta 6

5 / 5 pts

Dibuja el suffix trie de "araña". No es necesario entregues el dibujo.

¿Cuál(es) afirmaciones son verdaderas?

¡Correcto!

la raiz tiene 3 nodos hijos

verdadero



¡Correcto!

el nodo a que es hijo de la raíz tiene como etiquetas [0,2 4]]

¡Correcto!

tiene 15 nodos, sin contar la raíz

¡Correcto!

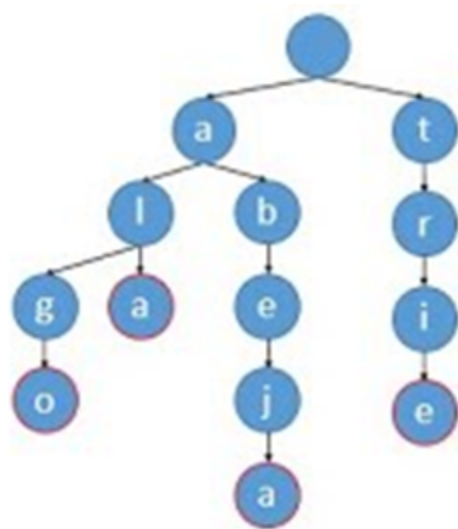
si buscamos "ra" no se encontrara, y devolvería Falso



Pregunta 7

5 / 5 pts

Considerando el siguiente árbol trie ¿cuál(es) de las siguientes afirmaciones son verdaderas?



¡Correcto!

no es un suffix trie

¡Correcto!

contiene 4 palabras

verdadero

▼

¡Correcto!
buscar si existe "abe" devolvería True

falso

▼

¡Correcto!
buscar si existe "algo" devolvería False

falso

▼

Otras opciones de coincidencia incorrecta:

- ninguno

⋮
Pregunta 8
5 / 5 pts

Considerando los caracteres con sus respectivas probabilidades

- A : 0.4
- B : 0.1
- E : 0.3
- C : 0.2

¿Cual es su árbol de Huffman? ¿Cual es el código de ACE?

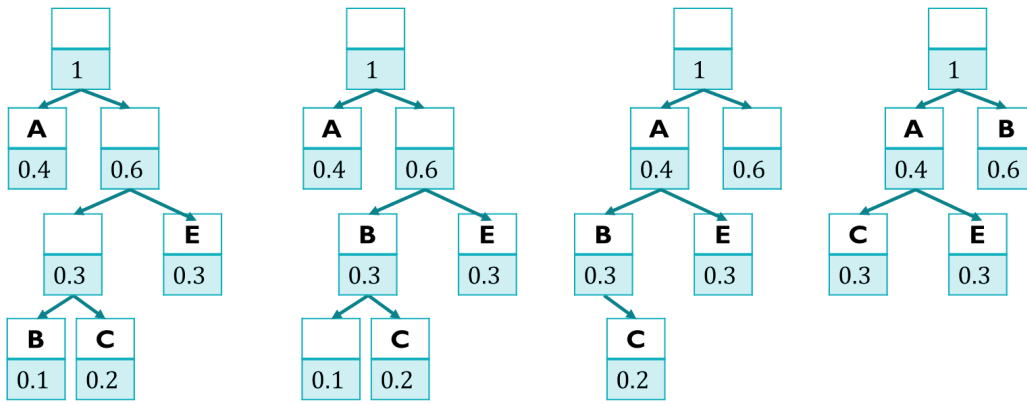
[Seleccionar]

▼

[Seleccionar]

▼

primero segundo tercero cuarto



Respuesta 1:

el tercero

¡Correcto!

el primero

el segundo

el cuarto

Respuesta 2:

000101

00100

¡Correcto!

010111

0101001

⋮

Pregunta 9

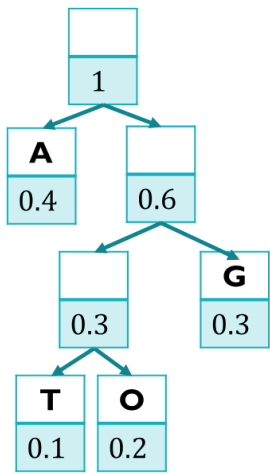
3.33 / 5 pts

Codifica usando el siguiente árbol de Huffman. Para que Canvas reconozca tu respuesta, usa un espacio para separar los codigos de cada caracter.

1) gato 11 0 100 101

2) toga 100 101 11 0

3) tatota 100 0 100 101 100



Respuesta 1:

¡Correcto! 11 0 100 101

Respuesta correcta

110100101

¡Correcto!

11 0 100 101

Respuesta 2:

¡Correcto! 100 101 11 0

¡Correcto!

100 101 11 0

Respuesta correcta

100101110

Respuesta 3:

Respondido 100 0 100 101 100 0

Respuesta correcta

100 0 100 101 0

Respuesta correcta

10001001010



Pregunta 10

0 / 5 pts

Completa correctamente:

Los arboles binarios de búsqueda son un ejemplo de la técnica de diseño divide y venceras .

Mas específicamente, operaciones de insertar y buscar son un ejemplo de decremento por factor constante .

Respuesta 1:

Respondido

divide y venceras

programación dinámica

fuerza bruta

Respuesta correcta

decrementa y venceras

Respuesta 2:

Respuesta correcta

decremento variable

Respondido

decremento por factor constante

tabulacion

incremento exponencial



Pregunta 11

5 / 5 pts

Asigna cada algoritmo con el tipo de técnica a la que pertenece.

¡Correcto!

bubble sort

fuerza bruta



¡Correcto!

merge sort

divide y venceras



¡Correcto!

quick select

decrementa y vencerás, cc



¡Correcto!

busqueda binaria

decrementa y vencerás, cc



Pregunta 12

5 / 5 pts

El problema resuelto en el siguiente pseudocódigo es , usando un enfoque de , cuya complejidad es $O(n^2)$

```
puntos = lista de puntos
n = cantidad de puntos

minimo = infinito
a = -1
b = -1

for i = 0 hasta n
    for j = i + 1 hasta n
        d = distancia de punto i a punto j
        si d < minimo:
            minimo = d
            a = i
            b = j

return punto a, punto b, j
```

Respuesta 1:

ruta mas corta que pasa por todos los puntos

¡Correcto!

closest pair

farthest pair

punto mas cercano a un punto dado

Respuesta 2:

¡Correcto!

fuerza bruta

búsqueda binaria

divide and conquer

programación dinámica

Respuesta 3:

¡Correcto!

$O(n^2)$

$O(\log n)$

$O(2n)$

$O(2^n)$



Pregunta 13

2.5 / 5 pts

Completa correctamente los siguientes enunciados, con base en las tres funciones a continuación.

En el grupo de la izquierda tienes las funciones en Python, y a la derecha en C/C++.

```
def fun_A(n):  
    if n == 0:  
        return 0  
    elif n == 1 :  
        return 1  
    else:  
        return funa_A(n-1) + fun_A(n-2)
```

```
def fun_B (n, a, b):  
    if n == 0:  
        return a  
    elif n == 1:  
        return b  
    else:  
        return fun_B (n-1, b, a+b )
```

```
def fun_C(n) :  
    serie = [0 for i in range(n+1)]  
  
    if n == 0:  
        return 0  
  
    serie[1] = 1  
    for i in range(2, n+1):  
        serie[i] = serie[i-1] + serie[i-2]  
  
    return serie[n]
```

```

int funA( int n )
{
    if (n == 0)
    { return 0; }
    else if (n == 1)
    { return 1; }
    else
    { return funA(n-1) + funA(n-2);}
}

int funB( int n, int a, int b )
{
    if (n == 0)
    { return a; }
    else if (n == 1)
    { return b; }
    else
    { return funB( n-1, b, a + b); }
}

int funC (int n)
{
    int i;
    int *serie = (int *) calloc(n , sizeof(int));

    if (n == 0)
    { return 0; }

    serie[1] = 1;
    for (i = 0; i < n ; i++)
    { serie[i] = serie[i-1] + serie[i-2]; }

    return serie[n];
}

```

¡Correcto!

fun_A

no usa programación dinar ▼

Respondido

fun_B

es un ejemplo de tabulación ✓

es un ejemplo de memoization

Respondido

fun_C

es un ejemplo de memoiza ✓

es un ejemplo de tabulación

¡Correcto!

la peor complejidad corresponde a

fun_A, que es exponencial ✓

Otras opciones de coincidencia incorrecta:

- calcula el factorial
- fun_C, que es cuadratica
- fun_B, que es exponencial
- es divide and conquer



Pregunta 14

0 / 5 pts

Considera la siguiente función para coin collecting (problema recolectar la maxima cantidad en monedas dispersas en una matriz, con movimientos solo hacia abajo y derecha)

La matriz de monedas se encuentra guardada en la variable llamada "matriz".

La función es una aplicación de:

F (i, j, matriz):

if i < 0 or j < 0:

return 0

else

if i >= 0 and j >= 0:

return max(F(i - 1, j, matriz), F(i, j - 1, matriz)) + matriz[i][j]

Respondido

- ☒ greedy
- ☐ programación dinámica
- ☐ tabulación

Respuesta correcta

- ☐ ninguna de las anteriores



Pregunta 15

2.5 / 5 pts

¿Cuáles de los siguientes tipos de algoritmos ofrecen soluciones optimas para problemas de optimización?

¡Correcto!

branch and bound

si

▼

Respondido

algoritmos greedy

si

▼

no

¡Correcto!

backtracking

no

▼

Respondido

busqueda exahustiva

no

▼

si



Pregunta 16

0 / 5 pts

Supongamos que debes resolver una instancia de knapsack de $n = 5000$ objetos. ¿Cuál algoritmo sería mas rápido y viable usar?

Respondido

☒ branch and bound

Respuesta correcta

☐ algoritmo greedy

☐ backtracking

☐ búsqueda exhaustiva



Pregunta 17

3.33 / 5 pts

¿Cuál es la versión en problema de decisión de TSP?

Encontrar un recorrido de costo minimo para pasar por todos los nodos y volver al origen

Maximizar el valor de una serie de objetos seleccionados sin sobrepasar la capacidad de un knapsack

Encontrar si hay un camino desde un nodo fijo hacia todos los demás

¡Correcto!

La version problema de decision de TSP es...

determinar si es posible re: ▼

¡Correcto!

En su version de optimización de TSP es un problema del tipo

optimización combinatoria ▼

Respondido

Un enfoque de fuerza bruta para TSP consistiría en

listar todos los posibles ca ▼

listar todas las posibles permutaciones

Otras opciones de coincidencia incorrecta:

- de factibilidad
- encontrar un recorrido de costo minimo para pasar por todos los nodos y volver al origen
- listar todos los posibles caminos de un nodo origen fijo al resto
- de optimización numérica y maximización
- encontrar si hay un camino desde un nodo fijo hacia todos los demás
- listar todos los posibles subconjuntos



Pregunta 18

4 / 5 pts

¿Cuáles niveles de complejidad se consideran intratables?

¡Correcto!

Exponenciales $O(2^n)$

¡Correcto!

Factoriales $O(n!)$

Respondido

polinómicos $O(n^m)$ con m constante

tratable

¡Correcto!

logarítmicos $O(n \log n)$

¡Correcto!

cuadráticos $O(n^2)$

tratable



Otras opciones de coincidencia incorrecta:

- ninguno



Pregunta 19

2 / 5 pts

Determina que tipo algoritmo es este, y completa los espacios en blanco.

[algoritmo] [A] [B] [C] [D]

```
La solución raíz entra en la fila
```

```
Mientras ____A____ :
```

```
    partial_sol = fila.pop()
```

```
    i = primer variable indefinida de partial_sol
```

```
    Por cada __B__ :
```

```
        Crea una solución hija
```

```
        Si la solución hija es valida
```

```
            Si esta completamente definida
```

```
                ____C____
```

```
            Si no
```

```
                ____D____
```

Respondido

el algoritmo es

branch and bound



backtracking

¡Correcto!

A

la fila no este vacia



¡Correcto!

B

valor en el dominio de la v



Respondido

C

comparar si es mejor que t



return hija

Respondido

D

return hija



hija entra a la fila

Otras opciones de coincidencia incorrecta:

- la fila este vacía
- calcular prioridad
- comparar si es mejor que best so far
- hija sale de la fila
- branch and bound
- branch and back
- variable indefinida



Pregunta 20

5 / 5 pts

¿Cuál de las siguientes afirmaciones **NO** es correcta sobre backtraking y branch and bound?

- ☐ las soluciones que sean invalidas se descartan
- ☐ ambos usan una fila para simular el árbol de soluciones
- ☐ tienen complejidades exponenciales / factoriales

¡Correcto!

☐ ambos resuelven problemas de optimizacion

Puntaje del examen: 55.16 de 100