Examen 02- Requires Respondus LockDown Browser

- Fecha de entrega 8 de nov en 13:30
- Puntos 100
- Preguntas 20
- Disponible después de 8 de nov en 11:00
- Límite de tiempo 90 minutos
- Es obligatorio utilizar el navegador Respondus LockDown Browser

Historial de intentos

	Intento	Hora	Puntaje
MÁS RECIENTE	Intento 1	69 minutos	75.5 de 100

Puntaje para este examen: 75.5 de 100
Entregado el 8 de nov en 12:25
Este intento tuvo una duración de 69 minutos.

Pregunta 1
0 / 5 pts

¿Que hace el siguiente algoritmo?

```
mistery(grafo G, nodo origen, nodo destino )

    stack.push(origen)
    Marcar origen como visitado
    predecesores = -1 para cada nodo

    Mientras stack no este vacía:
        u = stack.pop()
        Por cada nodo v vecino de u en G
            Si v no ha sido visitado
                stack.push(v)
                Marcar v como visitado
                predecesores[v] = u

                Si v == destino:
                  return predecesores
    return predecesores
```

- encontrar el camino mas corto de origen a destino usando el algoritmo de Dijsktra

Respuesta correcta

- encontrar un camino de origen a destino usando DFS
- recorrer todo el grafo en BFS

Respondido

- encontrar un camino de origen a destino usando BFS

Pregunta 2

0 / 5 pts

Supongamos que debes resolver una instancia de coloreado de grafos de n = 5000 objetos. ¿Cuál algoritmo de entre los siguientes seria mas rápido y viable usar?

Respondido

- programación dinámica
- branch and bound
- backtraking

Respuesta correcta

- algoritmo greedy

Pregunta 3

5 / 5 pts

Ayuda a Jhon Snow a pintar el siguiente diagrama de Voronoi de distintos colores. Jhon quiere que cada region tenga un color distinto a sus regiones vecinas.
Pero además, Jhon quiere gastar lo menos posible en pinturas.

Estos son los precios de las pinturas:

morado: $15

azul: $10

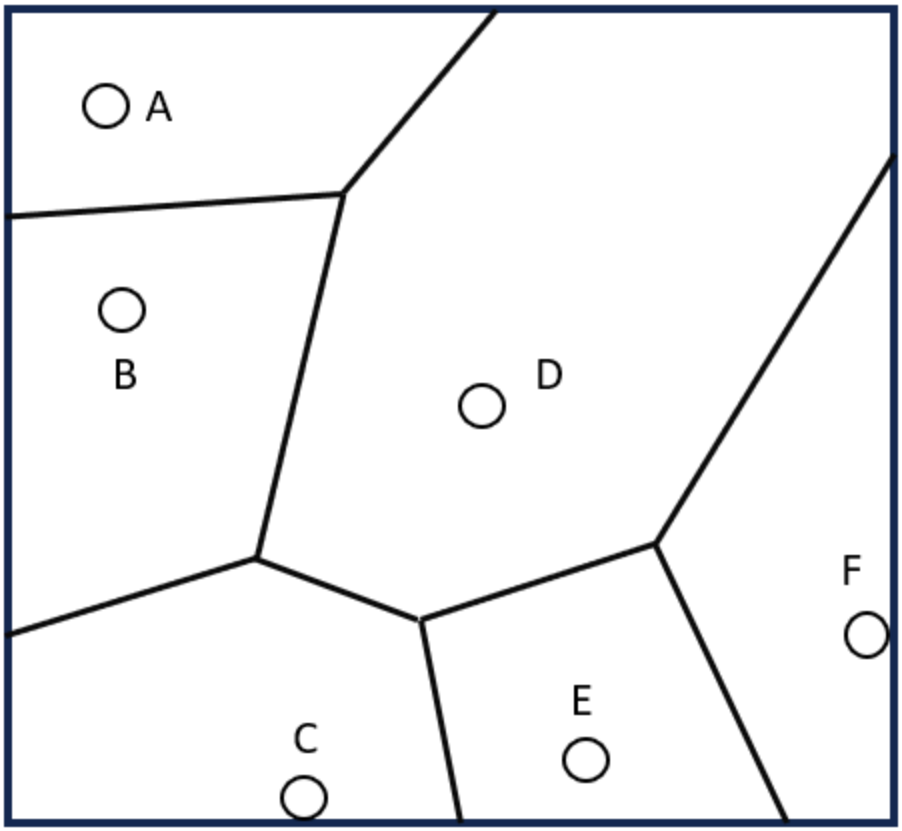
verde: $5

rosa: $12

amarillo: $11

rojo: $7

Describe en pseudocodigo como resolver el problema paso a paso, y la solución obtenida.
Especifica el color de cada area y el dinero gastado en total.



Su respuesta:

// ponemos los colores y guardaremos su repeticiones

Colores = [[color 1, rep = 0], [color2, rep = 0]...]

// reordenamos los nodos de nuestrografo en base a su grado y su color actual y los vecinos

Nodes = [[D, grado 5, color -1, vecinos [a,b,c,e,f]],[B,grado 3, color -1, vecinos[a,d,c]] ...]

Mientras no hayamos recorrido todos los nodos i del grafo

    //Marcamos el nodo como procesado

    Procesed [i] = true

    //Para el nodo actal buscamos el primer color disponible (que no repita con sus vecinos) comparando con los de susvecinos guardandolos en una cola de proiridad

    stack<nodos> vecinos = nodo->vecinos

    nodo -> color = buscarcolor(vecinos)

    // sumamos una repeticion a este color

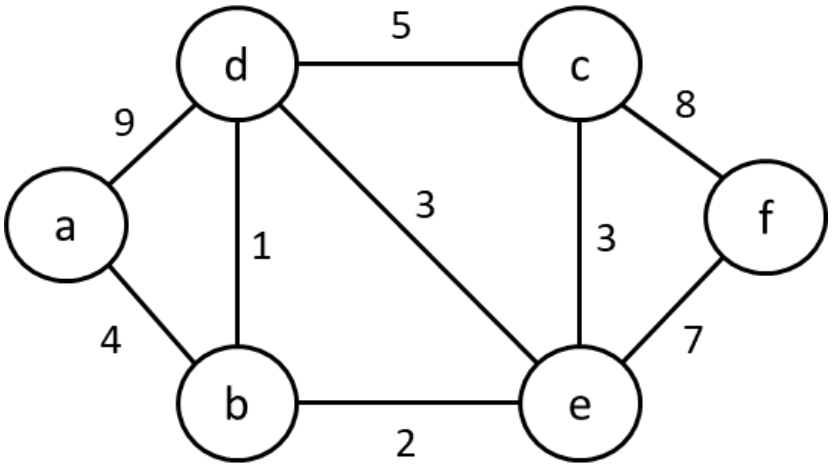
    colores[nodo-> color] ++

```
    // con este color se lo asignamos a todos los nodos que no sean vecinos directos de nuestro nodo
    // tambien comparamos con el maximo de colores y si es mayor lo guardamos

    nodo -> color > max ? max = nodo -> color

    //Regresamos los nodes


// reordenamos los colores en base a sus repeticiones

sort(colores , lambakey colores[1])

// asignamos estos colores con el nombre de las pinturas ordenadas en base a su costo

colores_nombres = [[verde,5],[rojo,7],[azul:10],[amarillo:11]..]

colores=[[color3,reps : 5] , [color 2, reps:3]...]

// mostramos los colores de cada region

para cada nodo i del grafo

    mostramos nodo -> color (como solo se guarda la posicion entonces ahora se mostrara el color
    nuevo que se aigno)

    // deberia quedar

    D -> azul

    B -> rojo

    E -> rojo

    A -> verde

    c -> verde

    E -> verde

//pasamos por los colores hasta max color y sumamos lo que costaria comprarlo una vez

para cada color i en colores hasta max

    suma += color_nombre[2]

// deberia quedar asi

suma = 22 (5+7+10)
```

Pregunta 4

5 / 5 pts

¿Cual es el spanning tree mínimo para el siguiente grafo?



- (a,b), (d,e), (b,e), (e, c), (e, f)

¡Correcto!

- (a,b), (d,b), (b,e), (e, c), (e, f)
- (a,b), (d,b), (d,c), (c, e), (e,f)
- (a,b), (b,d), (d,e), (e,c), (c, f)

Pregunta 5

5 / 5 pts

Si dos puntos A y B están conectados por un segmento en la triangulación de Delaunay ¿cual de las siguientes afirmaciones es correcta sobre A y B ?

¡Correcto!

- A y B comparten frontera en el diagrama de Voronoi
- A y B no comparten frontera en el diagrama de Voronoi
- El segmento que une a A y B en la triangulación es la frontera que separa sus zonas en el diagrama de Voronoi
- A y B no son parte del mismo triangulo en la triangulación de Delaunay

Pregunta 6

5 / 5 pts

¿Qué significa que el determinante de dos rectas resulte igual a 0?

- que las rectas son perpendiculares

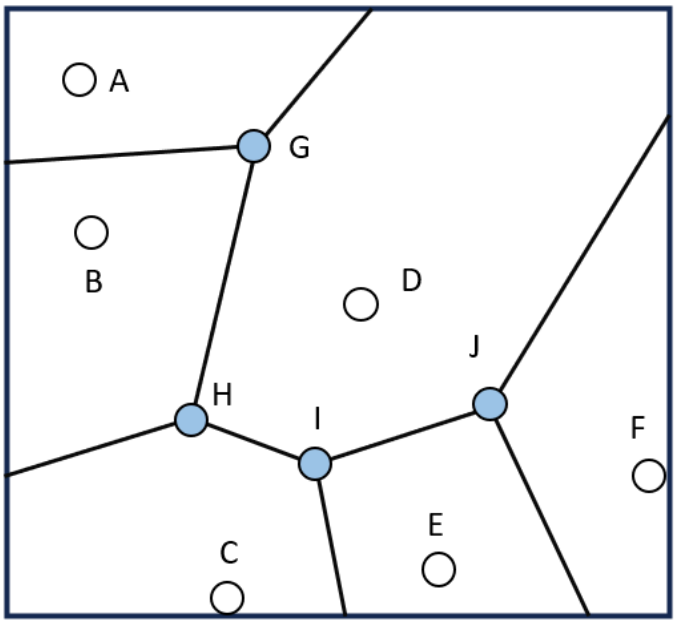
¡Correcto!

- las rectas son paralelas
- que las rectas se cruzan
- que las rectas se cruzan en dos puntos

Pregunta 7

5 / 5 pts

En el siguiente diagrama de Voronoi, los puntos G, H, I y J son...



- los vertices de los triángulos en la triangulación de Delaunay
- los puntos centrales de cada region

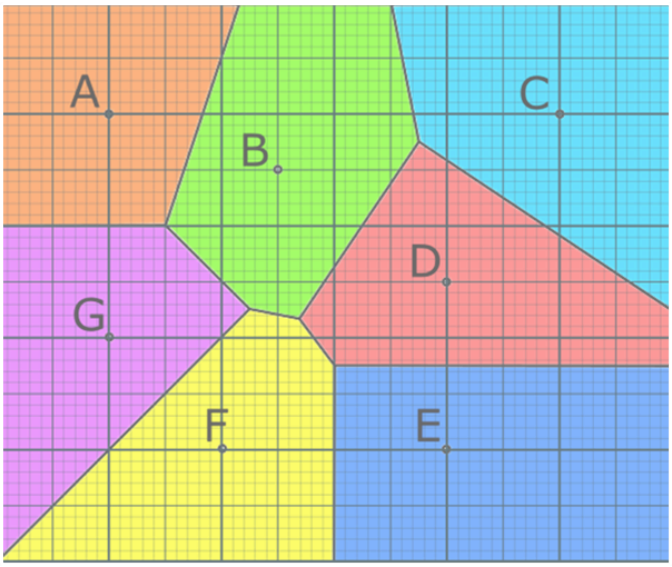
¡Correcto!

- los puntos de Voronoi, y circuncentros de los triángulos
- los puntos medios de los segmentos bisectores

Pregunta 8

5 / 5 pts

Considera el siguiente diagrama de Voronoi. ¿Cuáles puntos forman un triangulo en la triangulación de Delaunay? Marcalos como verdadero o falso.



¡Correcto!
B, C y D

verdadero

¡Correcto!
E, F, y D

verdadero

¡Correcto!
B, A y D

falso

¡Correcto!
G, F y D

falso

Pregunta 9
4 / 5 pts

El siguiente pseudocódigo corresponde al algoritmo para calcular el closest-pair de manera eficiente.

Rescribe en pseudocódigo su version iterativa equivalente.

OX = puntos ordenados según la x, OY = puntos ordenados según la y

findClosest(OX, OY)

n = número de puntos
If $n = 2$
return $d(OX[0], OX[1]),\ OX$
If $n = 3$
return $closest3(OX)$

$medio = \lceil n/2 \rceil$
$minL, closestL = findClosest(X[0 \ldots medio], OY)$
$minR, closestR = findClosest(X[medio + 1 \ldots n], OY)$

if $minL < minR$: $closestA = closestL$, $minA = minL$
else: $closestA = closestR$, $minA = minR$

S = elementos de OY cuya x este a distancia menor de $minA$ de $OX[medio]$
For cada i hasta $|S|$
For j de $i + 1$ a $\min(7, |S|)$
Si la distancia entre $S[i]$ y $S[j]$ es menor que $minA$, actualizar $minA$ y $closestA$

return $minA, closestA$

Su respuesta:

// Los puntos se ordenan igual

OX = puntos irdenaods , OY = puntos ordenados

findClosest(OX,OY)

// mismos casos base

n = numero de puntos

if n = retur distancia de los puntos

if n= 3 comparamos la tercia de puntos

// recorremos los puntos de mitada mitad (podemos usar 2 pointers)

for i= 1,j=medio+1 hasta i = medio

// iteramos sobre los otros puntos

For a desde i+1,j+1 hasta medio

if minL > d(Ox[i] , O[a]) actualizamos

if minR > d(Ox[j],Ox[a]) actualizamos

// comparambos ambas mitades

if minL < minR : closestA = closesL, minA=minL

else : closestA = closestR, minA = minR

S = elementos de OY cuya x este a distancia menor de minA en OX[medio]

for i hasta sizeof(S)

for jen i+1 hasta min(medio,sizeof(S))

si las distancias entre S[i] y S[j] < minA cambiamos minA y closestA

regresamos MinA y closestA

falto especificar el uso de pila

Pregunta 10

5 / 5 pts

El siguiente es el algoritmo de Floyd. ¿Que problema resuelve?

D = Matriz de adyacencia M, con 0's en la diagonal principal, e infinitos para nodos no-adyacentes

Para k = 0 hasta n

Para $i = 0$ hasta n

Para $j = 0$ hasta n

$$D[i,j] = \min(D[i,j], D[i,k] + D[k,j])$$

return la matriz de costos de los caminos D

- encontrar el camino mas corto hacia un destino especifico, desde todos los demas nodos
- encontrar si existe algun camino entre cada par de nodos, sin importar su longuitud
- encontrar el camino mas corto desde un unico nodo de origen hacia todos los demas

¡Correcto!

- encontrar el camino mas corto entre cada posible par de nodos

Pregunta 11

5 / 5 pts

¿Cuál de los siguientes puntos es correcto sobre la triangulación de Delaunay?

- Algunos puntos pueden quedar fuera de la triangulacion
- La salida es equivalente a un diagrama de Voronoi

¡Correcto!

- Para cada triangulo, su circunferencia circunscrita no contiene puntos
- Para cada triangulo, su circunferencia circunscrita contiene un punto en el centro

Pregunta 12

2.5 / 5 pts

El siguiente pseudocódigo corresponde al algoritmo para calcular el closest-pair de manera eficiente.

¿A cual tipo de técnica pertenece y cual es su complejidad?

OX = puntos ordenados según la x, OY = puntos ordenados según la y

findClosest(OX, OY)
- n = número de puntos
- If $n = 2$
  - return $d(OX[0], OX[1]),\ OX$
- If $n = 3$
  - return $closest3(OX)$

$medio = \lceil n/2 \rceil$
$minL, closestL = findClosest(X[0 \ldots medio], OY)$
$minR, closestR = findClosest(X[medio + 1 \ldots n], OY)$

if $minL < minR$: $closestA = closestL$, $minA = minL$
else: $closestA = closestR$, $minA = minR$

S = elementos de OY cuya x este a distancia menor de $minA$ de $OX[medio]$
For cada i hasta $|S|$
- For j de $i + 1$ a $\min(7, |S|)$
  - Si la distancia entre $S[i]$ y $S[j]$ es menor que $minA$, actualizar $minA$ y $closestA$

return $minA, closestA$

Respondido

Este algoritmo pertenece a la tecnica

programación dinámica

divide y venceras
¡Correcto!
Su complejidad es

O(n log n)

Otras opciones de coincidencia incorrecta:

- O(n^2)
- decrementa y vencerás
- programación dinámica
- O(7n log n)

Pregunta 13
0 / 5 pts

¿A cual clase de dificultad pertenece un problema con las siguientes caracteristicas?

-Es un problema de optimización, cuyo espacio de búsqueda crece exponencialmente

-Se cree que, en una maquina de Turing determinista (DTM), no hay un algoritmo polinómico que verifique la optimalidad de una solución

-Una maquina de Turing no-determinista (NDTM) podría resolverlo en tiempo polinómico.

-Se conoce un algoritmo polinómico (en DTM) que puede transformar este problema en cualquier otro problema NPO

Respondido

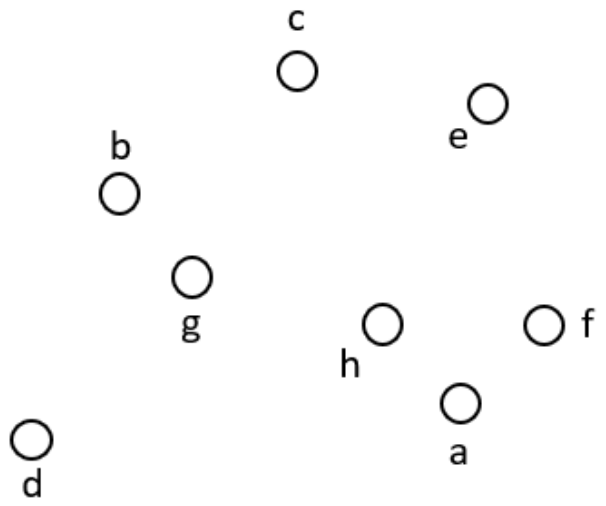
- (●) NP-Complete
- ( ) O(2^n)
- ( ) O(n!)

Respuesta correcta

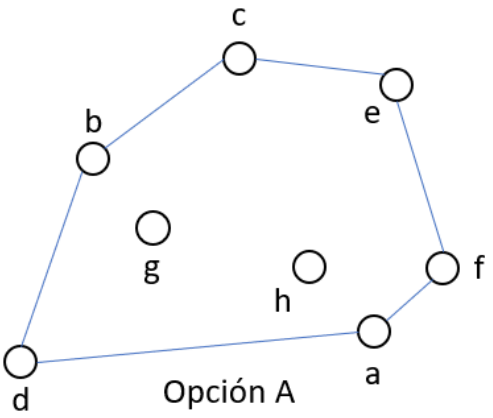
- ( ) NP-Hard

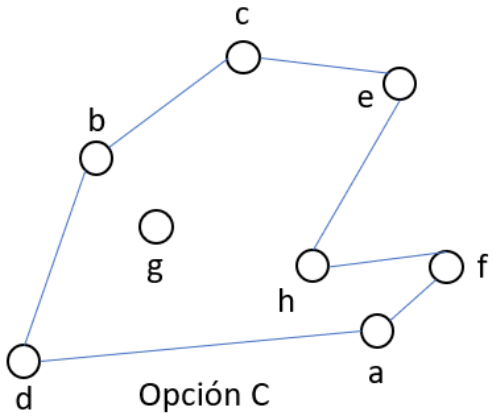
Pregunta 14
5 / 5 pts

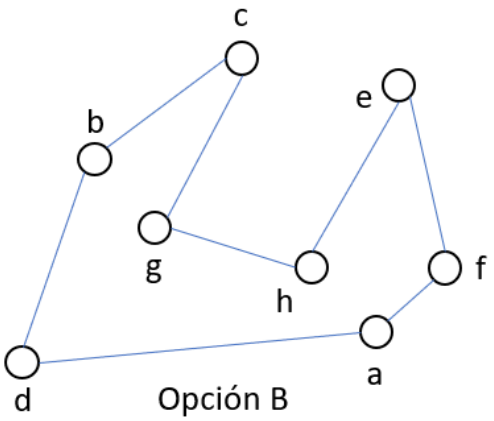
Dado el siguiente conjuntos de puntos ¿cual es su convex-hull?

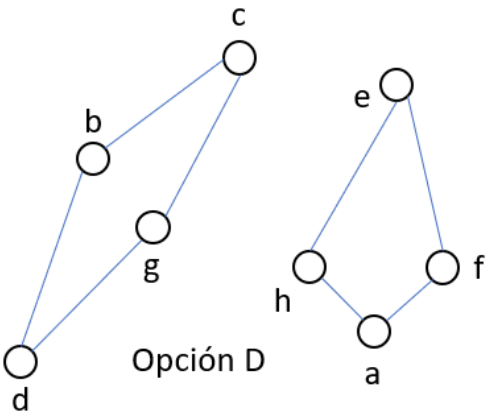


Opciones:









- C
- B

¡Correcto!

- A
- D

Pregunta 15
5 / 5 pts

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

A) Ana quiere representar el area de cobertura dominada por distintos supermercados, de modo que se pueda visualizar la region mas cercana a cada uno.

B) Karla debe conectar 5 sub-estaciones, de modo que formen una red conexa 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.

D) Marcos tiene un mapa de Mexico, y quiere pintarlo usando pocos colores sin que ningún par de estados vecinos sea del mismo color

¡Correcto!
Ana

Diagrama de Voronoi

¡Correcto!
Karla

Algoritmo de Prim

¡Correcto!
Lola

Algoritmo de ordenamiento

¡Correcto!
Marcos

Algoritmo Welsh-Powell

Otras opciones de coincidencia incorrecta:

- Algoritmo de Floyd

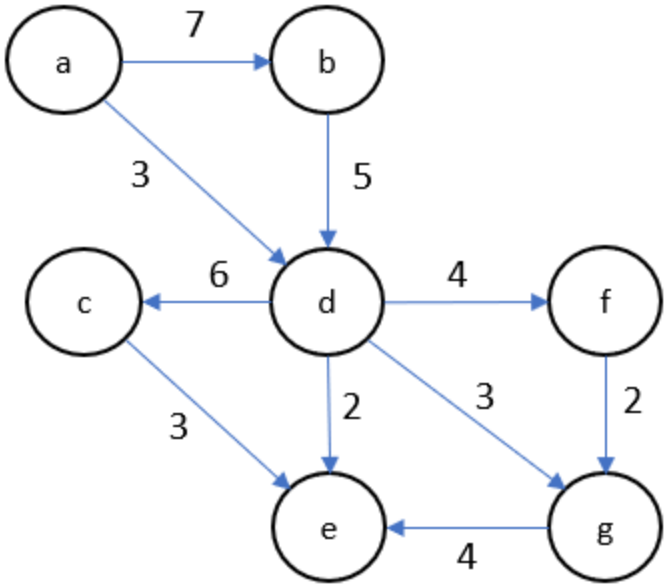
- Algoritmo de Dijkstra

Pregunta 16
5 / 5 pts

Calcula el maxFlow para el siguiente grafo.

Source = nodo a

Sink = nodo e



- 9
- 10

¡Correcto!

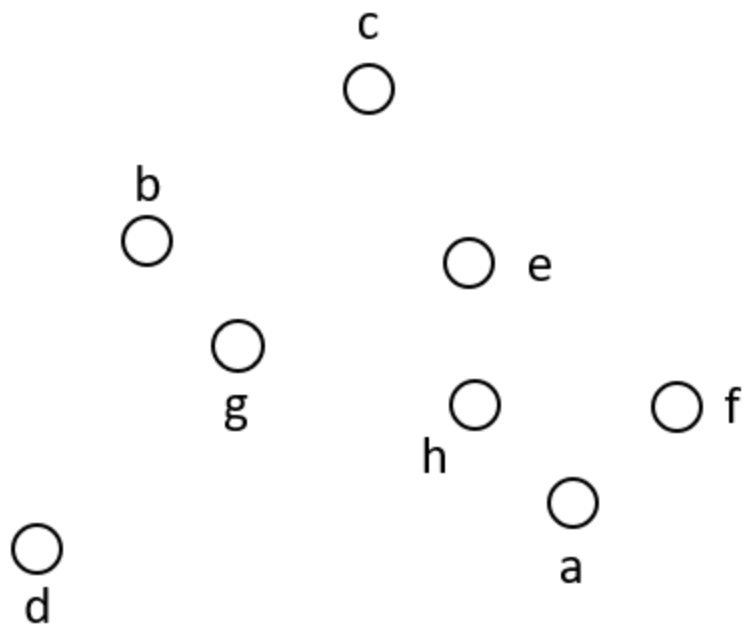
- 8
- 4

Pregunta 17
5 / 5 pts

Considera el problema de convex-hull para el siguiente conjunto de puntos y responde:

Si aplicamos el algoritmo Graham's scan, con d como punto de referencia, ¿en que orden se procesan los puntos? ¿que estructura se utiliza para procesarlos?

[Seleccionar] [Seleccionar]



Respuesta 1:

g, b, h, e, a, f, c

b, g, c,e, h, a, f

a, f,c, b

¡Correcto!

a, f, h, e, g, c, b

Respuesta 2:

arbol

fila de prioridad

fila

¡Correcto!

stack/pila

Pregunta 18

4 / 5 pts

¿Cuales de los siguientes algoritmos/estructuras sirven para resolver el problema de string matching?

¡Correcto!

Z-array

verdadero

Respondido

arreglo LPS

falso

verdadero
¡Correcto!
arbol de Huffman

falso

¡Correcto!
suffix trie

verdadero

¡Correcto!
trie tipo diccionario

falso

Pregunta 19
0 / 5 pts
Asigna cada algoritmo con el tipo de técnica a la que pertenece.
Respondido
Algoritmo Welsh-Powell

programación dinamica

Greedy
Respondido
Algoritmos Warshall-Floyd

fuerza bruta

programación dinamica

Respondido
Graham's scan

Greedy

decrementa y venceras, con decreemnto constante
Respondido
busqueda binaria

decrementa y venceras, cc

decrementa y vencerás, con factor constante = 2
Otras opciones de coincidencia incorrecta:

- divide y venceras
- fuerza bruta

Pregunta 20
5 / 5 pts
¿Cuáles de los siguientes tipos de algoritmos ofrecen soluciones optimas (en términos de calidad de solución) para problemas de optimización?
¡Correcto!
branch and bound

si

¡Correcto!
algoritmos greedy

no

¡Correcto!
backtracking

no

¡Correcto!

busqueda exahustiva

si

Puntaje del examen: 75.5 de 100