

# Examen-Diseno-De-Algoritmos-Febr...



**Apuntesgii**



**Diseño de Algoritmos**



**3º Grado en Ingeniería Informática**



**Escuela Superior de Ingeniería  
Universidad de Cádiz**

Formamos  
**talento** para un futuro  
**Sostenible**



MÁSTER EN

**Big Data &  
Business Analytics**

**EOI** Escuela de  
organización  
industrial

[saber más](#)

Lo que te pide esta cuenta es lo mismo  
que hiciste el finde que dijiste que te  
ibas a poner al día NADA.



por BANCA MEXICO en cumplimiento de la Ley del Sistema de Pagos de Depósitos. No se garantiza el pago de depósitos. Consulta más información en [legas.es](http://legas.es)

1/6  
Este número es el indicador del riesgo del producto, siendo 1 el indicador de menor riesgo y 6 el de mayor riesgo.

¡Píllala aquí!

\*TIN 0 % y TAE 0 %.

Cuenta NoCuenta, la cuenta sin comisiones\* que no te pide nada.

## Examen Diseño de Algoritmos, Febrero 2023 y su solución

Ejercicio 1. Se dispone de un buque portacontenedores que hay que cargar con tantos contenedores disponibles en la terminal de carga como sea posible, todos de idénticas dimensiones, pero cada uno con su propio peso.

El buque podría cargar todos los contenedores de estar vacíos, pero al no ser el caso, su peso total podría exceder la capacidad de carga del buque, lo que no debe suceder.

Diseñe un algoritmo devorador para intentar resolver el problema. Mejore su eficiencia con preordenación, y luego, con montículo.

- Identificar los elementos del esquema
- Diseñar el algoritmo
- Analizar la eficiencia
- Explicar cómo mejorarlo con las técnicas y analizar las mejoras

Ejercicio 2. Una empresa dispone de una serie de centros logísticos, una flota de vehículos permite transportar paquetes entre ellos, para lo que la empresa ha establecido un mapa con las rutas que los conectan.

Al realizar cualquier ruta entre dos centros logísticos, el vehículo asignado debe parar en todos los centros intermedios situados en ella, de forma que pueda procederse a la carga y descarga de los paquetes necesarios.

Diseñe un algoritmo que determine cuántas paradas ha de realizar como MINIMO un vehículo en ruta entre los dos centros más alejados de la red antes de llegar a su destino.

- Diseñar tabla de subproblemas resueltos.
- Diseñar algoritmo.
- Analizar eficiencia temporal y espacial del algoritmo.



do your thing

WUOLAH  
Apuntes GII

Ejercicio 3. Sea  $v = [v_1, \dots, v_n] \in \mathbb{Z}^+$  tal que  $v_1 < \dots < v_n$ . Diseñe un algoritmo de coste logarítmico que permita encontrar un índice  $p \in [1, N]$  tal que  $vp = p$ , si existe un índice tal, o bien devuelva 0 en caso de que no exista

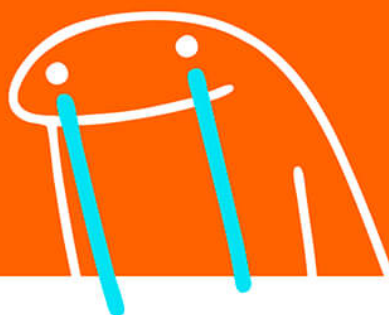
- Identificar elementos del esquema.
- Diseñar el algoritmo correctamente con su eficiencia.
- Plantear ecuación de recurrencia y resolverla.

Ejercicio 4. Diseñe un algoritmo que resuelva un laberinto tridimensional mediante búsqueda con retroceso. El laberinto está representado por un cubo en el que, originalmente, cada celda aparece libre (' ') o bloqueada ('X'). Se trata de encontrar, si es posible, una salida desde una posición inicial dada. A este efecto, se considera que una salida es cualquier celda libre del borde, es decir, de una de las caras del cubo. El algoritmo recibirá el laberinto y la posición inicial, que ha de ser válida y corresponder a una celda libre, y devolverá un booleano que indique si existe o no solución. Además, debe marcar las celdas visitadas ('V') y las que forman parte del camino de salida ('S'), si este existe.

- Diseñe un algoritmo que resuelva el problema.
- Explicar la complejidad temporal.

Si ya tuviste sufi con tanto estudio...  
Te dejamos este espacio  
para desahogarte.

Pinta, arranca,  
llora... tú decides ;)



¿Te sientes más liberado?  
Sigue siéndolo con la **Cuenta NoCuenta:**  
**libre de comisiones\*, y de lloraditas.**

**¡Quiero una de esas!**

\*TIN 0 % y TAE 0 %.



do your thing

# Diseño de Algoritmos



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**



**Banco de apuntes de la**

**WUOLAH**

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



## Nota previa a las soluciones:

Hola, soy un estudiante como tú que ha tenido que estudiar esta asignatura. Este examen específico, lo he realizado mientras preparaba la asignatura y aunque **pienso** que las soluciones propuestas son correctas, quiero que entiendas que a lo mejor estas no son las mejores, o a lo mejor no son del todo válidas. Si ves algún fallo, te animo a dejarlo en los comentarios del documento, así todos podemos observarlo y tenerlo en cuenta.

Échale codos, tú puedes. :)

Soluciones:

### • Ejercicio nº1

- a) - Conjunto de candidatos: buques  
- Conjunto de candidatos seleccionados: S  
- Función de factibilidad: Si incluye el buque supone al peso max?  
- Función solución: no es necesario (se ha llegado al peso max?)  
- Función de selección: buque de menor peso.  
- Función objetivo: número de buques  
- Objetivo: maximizar.

b) El algoritmo tiene un coste en el peor de los casos de  $O(n^2)$

Lista con los contenedores (Representados por sus pesos)

b)  $\text{Cargar Buque}(\text{Contenedores}, \text{CargaMaxima}) \rightarrow S$   
 $C_1 \leftarrow \text{Contenedores}$   
 $S \leftarrow \emptyset$   
 $\text{CargaActual} \leftarrow 0$   
mientras  $\text{CargaActual} < \text{CargaMaxima} \wedge C_1 \neq \emptyset$   
     $\text{ContenedorActual} \leftarrow \text{SeleccionarMenor}(C_1)$   
     $C_1 \leftarrow C_1 - \{\text{ContenedorActual}\}$   
    si  $\text{CargaActual} + \text{ContenedorActual} \leq \text{CargaMaxima}$   
         $\text{CargaActual} \leftarrow \text{CargaActual} + \text{ContenedorActual}$   
         $S \leftarrow S \cup \{\text{ContenedorActual}\}$

$\text{SeleccionarMenor}(C_1) \rightarrow v$   
 $v \leftarrow \infty$   
para todo  $\text{Contenedor} \in C_1$   
    si  $\text{Contenedor} < v$   
         $v \leftarrow \text{Contenedor}$

c)  $\text{Cargar Buque}(\text{Contenedores}, \text{CargaMaxima}) \rightarrow S$   
 $C_1 \leftarrow \text{Contenedores}$   
 $S \leftarrow \emptyset$   
 $\text{CargaActual} \leftarrow 0$   
 $\text{Ordenar}(C_1)$   
mientras  $\text{CargaActual} < \text{CargaMaxima} \wedge C_1 \neq \emptyset$   
     $\text{ContenedorActual} \leftarrow \text{Extraer-Primeros}(C_1)$   
    si  $\text{CargaActual} + \text{ContenedorActual} \leq \text{CargaMaxima}$   
         $\text{CargaActual} \leftarrow \text{CargaActual} + \text{ContenedorActual}$   
         $S \leftarrow S \cup \{\text{ContenedorActual}\}$

$\text{Cargar Buque}(\text{Contenedores}, \text{CargaMaxima}) \rightarrow S$   
 $C_1 \leftarrow \text{Monticulo}(\text{Contenedores})$   
 $S \leftarrow \emptyset$   
 $\text{CargaActual} \leftarrow 0$   
mientras  $\text{CargaActual} < \text{CargaMaxima} \wedge C_1 \neq \emptyset$   
     $\text{ContenedorActual} \leftarrow \text{Raiz}(C_1)$   
    Eliminar Raiz( $C_1$ )  
    si  $\text{CargaActual} + \text{ContenedorActual} \leq \text{CargaMaxima}$   
         $\text{CargaActual} \leftarrow \text{CargaActual} + \text{ContenedorActual}$   
         $S \leftarrow S \cup \{\text{ContenedorActual}\}$

Las mejoras son notables ya que en ambos casos pasamos a tener un coste en el peor de los casos de  $O(n \cdot \log n)$



Lo que te pide esta cuenta es lo mismo que hiciste el finde que dijiste que te ibas a poner al día NADA.

Por SANE ME se encuentra adherido al Sistema de Garantía de Depósitos. Manténlo con una garantía de hasta 200.000 euros por depositante. Consulta más información en [lag.es](http://lag.es)

1/6

Este número es indicativo del riesgo del producto, siendo 1 el indicativo de menor riesgo y 6 el de mayor riesgo.

¡Píllala aquí!

TIN 0 % y TAE 0 %.



### Ejercicio n°2:

- a) La tabla de subproblemas resueltos sera una matriz de coste donde se indicará el mínimo de centros logísticos entre dos centros logísticos
- c) El coste temporal del algoritmo es de  $t(n) \in O(n^3)$   
El coste espacial del algoritmo es de  $t(n) \in O(n^2)$

matriz de Adyacencia

b) Mínimo Paredes  $(a, n) \rightarrow c$

desde  $i \leftarrow 1$  hasta  $n$   
desde  $j \leftarrow 1$  hasta  $n$   
si  $a[i, j]$   
     $r[i, j] \leftarrow 1$   
sino  
     $r[i, j] \leftarrow \infty$

Floyd  $(r, n)$   
 $c \leftarrow -\infty$   
desde  $i \leftarrow 1$  hasta  $n$   
desde  $j \leftarrow 1$  hasta  $n$   
si  $r[i, j] > v$   
     $c \leftarrow r[i, j]$

Floyd  $(r, n) \rightarrow r$   
desde  $i \leftarrow 1$  hasta  $n$   
 $r[i, i] \leftarrow 0$   
desde  $k \leftarrow 1$  hasta  $n$   
desde  $i \leftarrow 1$  hasta  $n$   
desde  $j \leftarrow 1$  hasta  $n$   
 $r[i, j] \leftarrow \min(r[i, j], r[i, k] + r[k, j])$

### Ejercicio n°3:

- a) Descomposición: El problema se divide en subproblemas de menor tamaño.
- Resolución Recursiva: El problema se resuelve en contextos de recursividad.
  - Combinación: la solución se compone de las soluciones de los subproblemas.
  - Subesquema: dado a que el problema se divide en un subproblema de menor tamaño, pertenece a la simplificación.

c)

$t(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ t(n-2) + 2 & \text{si } n \geq 2 \end{cases}$

No homogénea

$\hookrightarrow T_{\text{homogenea}} \rightarrow T(k) = T(k-1) + 2 \rightarrow T(k) - T(k-1) = 2$

$C_1(x) = (x-1)$   
 $C_2(x) \Rightarrow h(n) = 2 \rightarrow h(n) = 2 \cdot 1^k \cdot k^0 \begin{cases} s_1 = 1 \\ m_1 = 0 \end{cases}$   
 $\hookrightarrow (x-1)^2$

$C(x) = (x-1)^2 \rightarrow B = \{1, k\}$

$T(k) = h_1 + h_2 k \rightarrow T(0) = h_1 + h_2 \cdot 0 = 1 \rightarrow \begin{cases} h_1 = 1 \\ h_2 = 2 \end{cases}$

si  $n = 1 \rightarrow t(n) = 1$   
si  $2^k = 1$   
si  $k = 0$

$T(k) = 2k + 1 \rightarrow t(n) = 2 \cdot \log_2(n) + 1 \rightarrow t(n) \in O(\log(n))$   
 $n = 2^k \rightarrow k = \log_2(n)$

b) Búsqueda Binaria Indice  $(v, i, j) \rightarrow r$

$n \leftarrow j - i + 1$   
si  $n = 1$   
    si  $v[i] = i$   
         $r \leftarrow i$   
    sino  
         $r \leftarrow 0$   
 $k \leftarrow i - 1 + n \text{ div } 2$   
si  $v[k] = k$   
     $r \leftarrow k$   
sino  
    si  $v[k] < k$   
        Búsqueda Binaria Indice  $(v, i, k)$   
    sino  
        Búsqueda Binaria Indice  $(v, k, j)$

### Ejercicio n°4

- b) La complejidad temporal es de  $t(n) \in O(m \cdot n)$

a) SolucionLubeninto  $(l, m, n, z, i, j, k) \rightarrow (l, s)$

si  $l[i, j, z] = ' '$   
    si  $\text{bonde}(m, n, z, i, j, k)$   
         $s \leftarrow T$   
    si no  
         $l[i, j, z] \leftarrow 'v'$   
         $s \leftarrow \text{SolucionLubeninto}(l, m, n, z, i-1, j, k) \vee$   
             $\text{SolucionLubeninto}(l, m, n, z, i+1, j, k) \vee$   
             $\text{SolucionLubeninto}(l, m, n, z, i, j-1, k) \vee$   
             $\text{SolucionLubeninto}(l, m, n, z, i, j+1, k) \vee$   
             $\text{SolucionLubeninto}(l, m, n, z, i, j, k-1) \vee$   
             $\text{SolucionLubeninto}(l, m, n, z, i, j, k+1)$   
    si  $s$   
         $l[i, j, z] \leftarrow 's'$   
si no  
     $s \leftarrow \perp$

Apuntes GII

Cuenta NoCuenta, la cuenta sin comisiones\* que no te pide nada.



do your thing