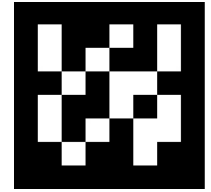


--	--	--	--	--	--	--	--	--	--

Matricula



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PRIMERA EVALUACIÓN – I TÉRMINO 2016-2017 DISEÑO Y ANALISIS DE ALGORITMOS

Nombre: _____

Instrucciones: **Ponga el número de matrícula en TODAS las hoja y solamente conteste en las hojas suministradas**

Pregunta 1 (15 puntos)

Considere el siguiente problema: Tiene una matriz de $n \times n$ $A[i, j]$, $1 \leq i, j \leq n$. Suponga que cada fila y cada columna de A están ordenadas, por ejemplo: para una i fija, $A[i, j]$ incrementa con j y para una j fija incrementa con i .

Diseñe un algoritmo que haga lo siguiente

Entrada: un número x .

Salida: Verdadero si x está en A , Falso sino.

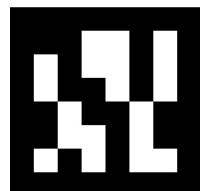
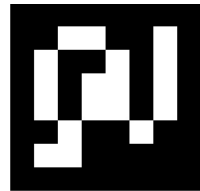
Diseñe el algoritmo a través de la técnica de Dividir y Conquistar. El rendimiento del algoritmo debe ser mejor que $O(n^2)$

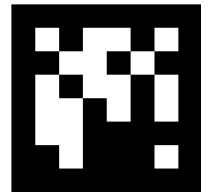
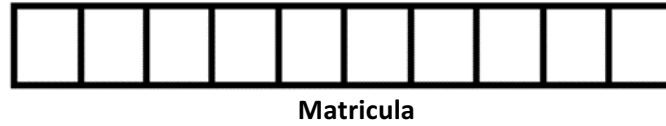
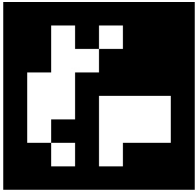




--	--	--	--	--	--	--	--	--	--	--

Matricula

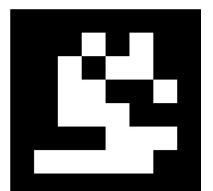
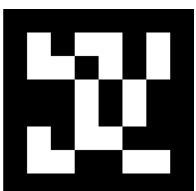


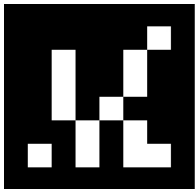


Pregunta 2 (10 puntos)

Le dan una función CombineSort recursiva que ordena una entrada $A=\{a_1, a_2, \dots, a_n\}$. Encuentre la relación de recurrencia $T(n)$ del algoritmo y úsela para calcular el tiempo asintótico de ejecución del algoritmo en el peor de los casos (Nota: la función Combine se ejecuta en $O(n)$ en el peor caso).

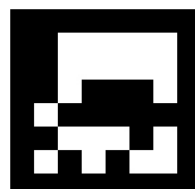
```
Start = 1;
Finish = n;
BEGIN CombineSort(A, Start, Finish)
    IF (Start < Finish) THEN
        Quarter=(Start + Finish)/4;
        CombineSort(A, Start, Quarter);
        CombineSort(A, Quarter+1, 2*Quarter);
        CombineSort(A, 2*Quarter+1, 3*Quarter);
        CombineSort(A, 3*Quarter+1, Finish);
        Combine(A, Start, Quarter, Finish);
    END IF
END BEGIN
```

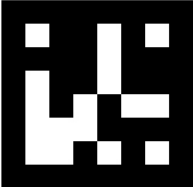




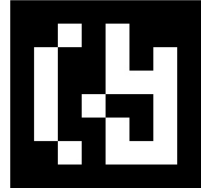
--	--	--	--	--	--	--	--	--	--	--

Matricula





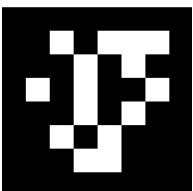
Matricula

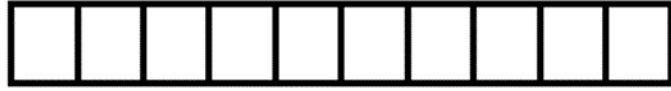


Pregunta 3 (15 puntos)

Considere un Min-Heap representado por un arreglo $A[1..n]$.

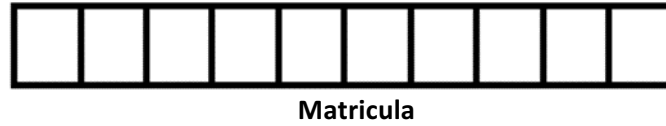
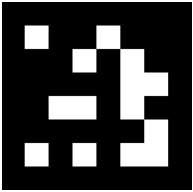
- a) Suponga que necesita cambiar el valor de el i -ésimo element. Cree un algoritmo eficiente que al cambiar $A[i]=x$ restaure la propiedad heap.
- b) Analice el algoritmo propuesto y encuentre el límite asintótico del tiempo de ejecución del algoritmo en el peor de los casos.





Matricula





Pregunta 4 (10 puntos)

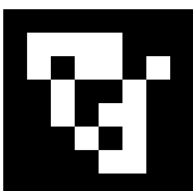
Dado el siguiente algoritmo

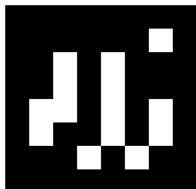
```
for i ← 1 to 100 do
  for k ← 1 to n do
    j ← 1 ; m ← n
    while j < m do
      m ← (m + j) / 2
    end while
  end for
end for

for i ← 1 to 30 do
  for j ← 1 to n do
    k ← i + j + n
  end for
end for

for i ← 1 to 70 do
  j ← 2*n + i
end for
```

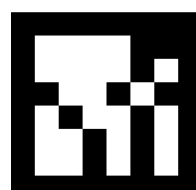
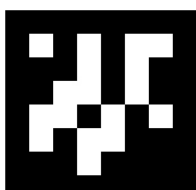
- ¿Cuál es el tiempo exacto de ejecución con respecto a n ?
- ¿Cuál es el tiempo de ejecución en notación O ? Pruébalo usando la definición matemática
- ¿Cuál es el tiempo de ejecución en notación Omega mayúscula? Pruébalo usando la definición matemática

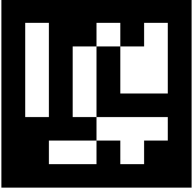




--	--	--	--	--	--	--	--	--	--	--

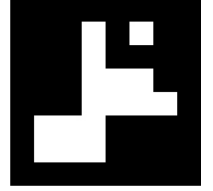
Matricula





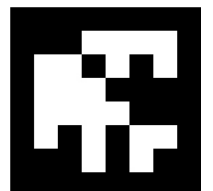
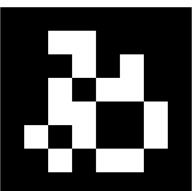
--	--	--	--	--	--	--	--	--	--

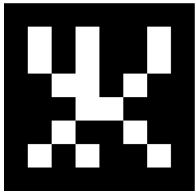
Matricula



Pregunta 5 (5 puntos extra)

Describe 3 aspectos del curso que podrían ser mejorados y como, en su criterio, se deberían mejorar.





--	--	--	--	--	--	--	--	--	--

Matricula

