DÍA 6 (5) CLASE MIERCOLES 05-MAYO-2021

INICIO DE CLASE-

1) Crear algoritmos para determinar si un número ingresado es par o impar

Inicio
Print “ingrese un número”
Leer un número
Si (número ingresado%2==0) es par
Si (número ingresado%2!==0) es impar
Print resultado de operación
Fin

2) Crear algoritmos para determinar si un número ingresado es múltiplo o no de 2

Inicio
Print “ingrese un número”
Leer un número
Si (número ingresado%2==0) es múltiplo de 2
Si (número ingresado%2!==0) no es múliplo de 2
Print resultado de operación
Fin

Problemas aquí… ¿Está bien? Me confundí un montón…

3) Crear algoritmos para determinar si un número ingresado es múltiplo o no de 3

Inicio
Print “ingrese un número”
Leer un número
Si (número ingresado%2==0) no es múltiplo de 3
Si (número ingresado%2!==0) es
Print resultado de operación
Fin

¿? Tengo que volver a revisar esto

4) Crear algoritmos que permita determinar cuál es el mayor de los dos números ingresados

Inicio
Print “ingrese un primer número”
Leer primer número
Print “ingrese un segúndo número”
Leer segundo número
Si (primer número > segundo número)
Print primer número > segundo número
Si (primer número < segundo número)
Print primer número < segundo número
Fin

5) Crear algoritmos que permita determinar si la suma de dos números ingresados es positiva, negativa o cero

Inicio
Print “ingrese un primer número”, a
Leer a
Print “ingrese un segúndo número”, b
Leer b
Si a+b>0
Print a+b>0
Si a+b==0
Print a+b=0
Si a+b<0
Print a+b<0
Fin

También tengo que acostumbrarme a diferencias esto = de esto ==, porque los confundo

6) Crear algoritmos que permita determinar si un número es divisible por 2 y por 5 al mismo tiempo

Inicio
Print “ingrese un número”, a
Leer a
Si a%10==0
Print “a es divisible por 2 y por 5 a la vez”
Fin

Aún no me acostumbro a la operación de %, es la segunda vez que la veo, debo trabajarla.

7) Crear algoritmos que permita determinar cuál es el mayor de tres números ingresados

Inicio
Print “ingrese un primer número”, a
Leer a
Print “ingrese un segúndo número”, b
Leer b
Print “ingrese un tercer número”, c
Leer c
Si c<a>b
Print “a es mayor”
Si c<b>a
Print “b es mayor”
Si a<c>b
Print “b es mayor”
Fin

8) Crear algoritmos que solicite al usuario ingresar dos números y realizar la operación matemática con estos números, el usuario debe seleccionar la operación a realizar

Inicio
Sumar = +
Restar = -
Multiplicar = *
Dividir = /
Print “ingrese un número”, a
Leer a
Print “ingrese la operación matemática que desea realizar”, + o – o * o /
Leer +
Print “ingrese un número”, b
Leer b
Print “a+b”
Fin

Este ejercicio me complicó ☹ como que me faltan pasos parece, creo que me comí algo importante, en la parte de poner las operaciones matemáticas pero no sé qué es…

Estructura iterativas: tipo de estructuras que nos permiten realizar repeticiones de nuestras acciones, de manera que cada acción se realiza varias veces.

Mientras: es un bucle que realiza repetición de las instrucciones hasta que deje de cumplirse una cierta condición (mientras la condición se cumpla, se repite el ciclo)

9) Realizar un algoritmo en pseudocódigo que me muestre por pantalla los números del 1 al 5.

Inicio
a=1 *(en la segunda vuelta, mi variable a será 2, en la tercera 3, y así)*
Mientras a<=5 hacer
Escribir a
a=a+1 *(en este paso lo que hago es otorgarle nuevo valor a mi variable a)*
Fin mientras

REPASAR. Tengo que pensarlo lento para entenderlo bien.

Este ciclo no nos asegura una ejecución, ya que puede que la condición de mientras nunca se cumpla.

Hacer mientras: cumple la misma función que la estructura mientras, es un ciclo repetitivo. La diferencia está en que Hacer mientras **nos asegura la ejecución de la instrucción (una vez)** y luego verifica si se cumple o no la condición.

Inicio
PARA (a=valor inicial; a<valor final; a++) HACER
Instrucciones
Fin

REPASAR. Tengo que pensarlo lento para entenderlo bien.

Para: realiza repeticiones , pero **definidos** en un intervalo (**desde un valor inicial hasta un valor final**) y se indica como es el incremento o decremento de dicho valor.

```
INICIO
PARA (a=1; a<=5; a++) HACER
        Escribir a
FIN PARA
```

REPASAR. Tengo que pensarlo lento para entenderlo bien.

a++ = a + 1

a - - = a - 1

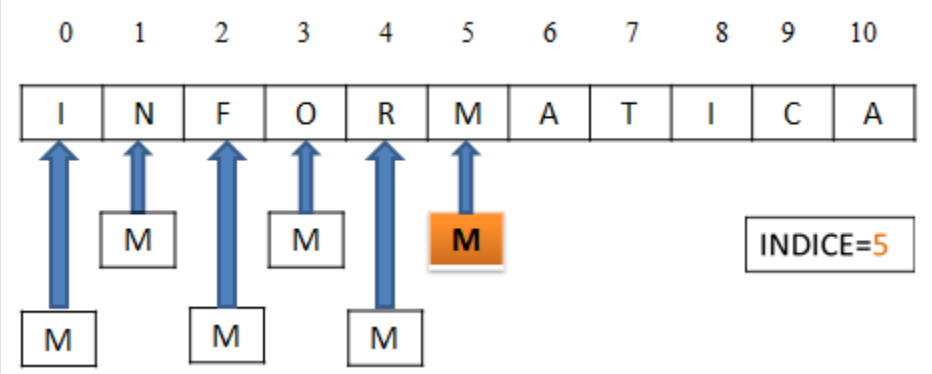
Si quiero incrementar de 2 en 2, lo escribo como 2 + +

Si quiero decrementar de 5 en 5, lo escribo como 5 - -

10) Buscar información sobre los algoritmos de búsqueda y ordenamiento

ALGORITMOS DE BÚSQUEDA

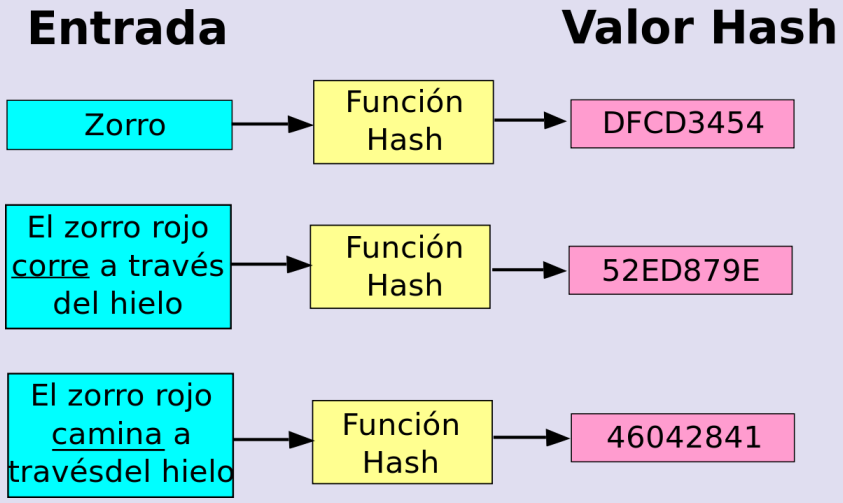
- **Búsqueda Secuencial**: Consiste en ir comparando el elemento que se busca con cada elemento del arreglo hasta que se encuentra (*como lo que hacían algunos programas que buscaban contraseñas)*



- **Búsqueda Binaria**: Para utilizar este algoritmo, el arreglo debe estar ordenado y no se deben repetir los elementos. La búsqueda binaria consiste en dividir el arreglo en dos subarreglos más pequeños, y comparar el elemento con el del centro. Si coinciden, la búsqueda se termina. En cada iteración el arreglo se divide en dos.

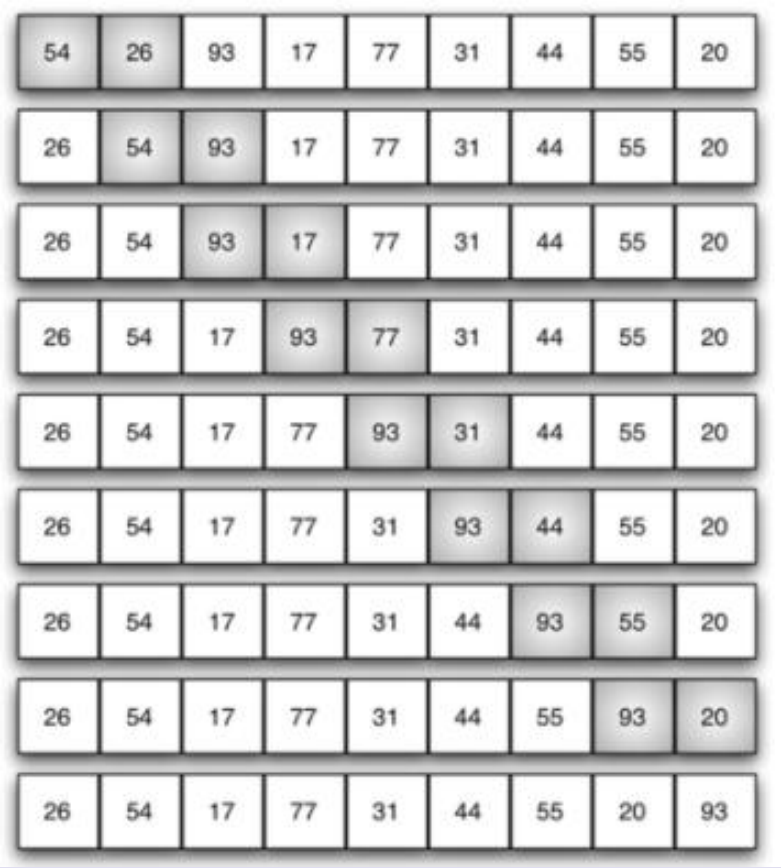
KHÉ? XD

- **Búsqueda Hashing**: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (*casilla* o *cubeta*) donde la tabla hash localiza el valor deseado.

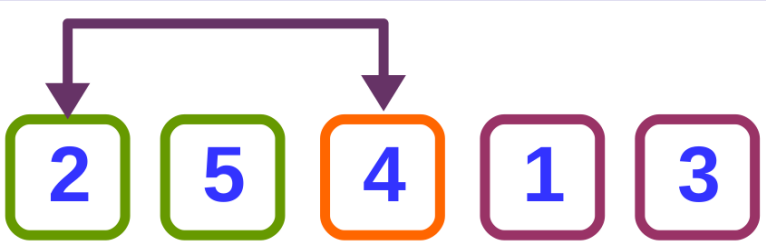


ALGORITMOS DE ORDENAMIENTO

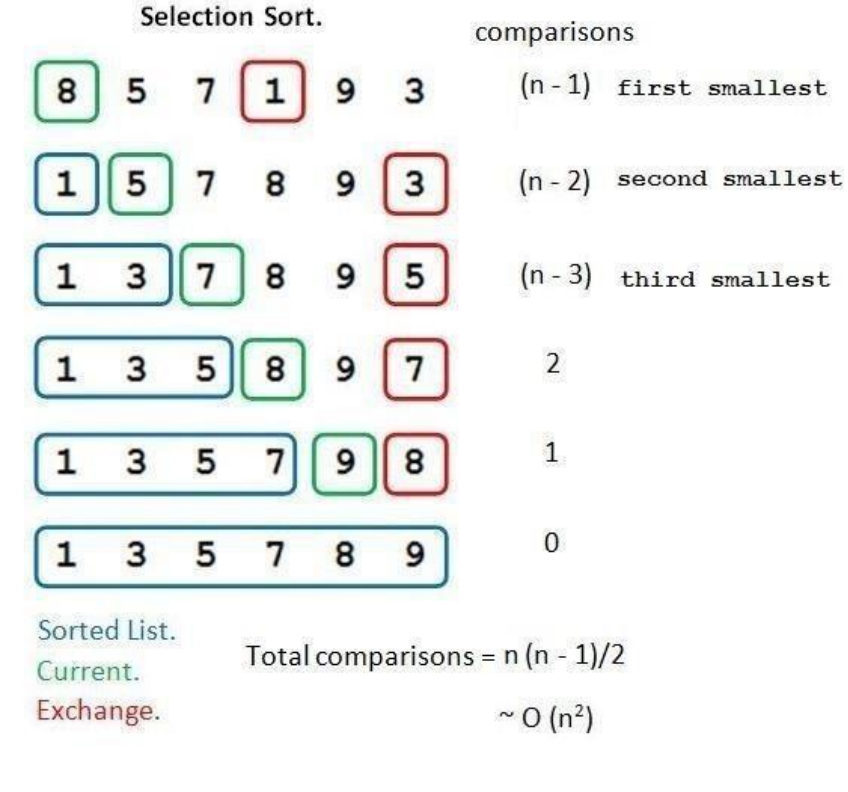
- **Ordenamiento de Burbuja**: Consiste en ciclar repetidamente a través de la lista, comparando elementos adyacentes de dos en dos. Si un elemento es mayor que el que está en la siguiente posición se intercambian.



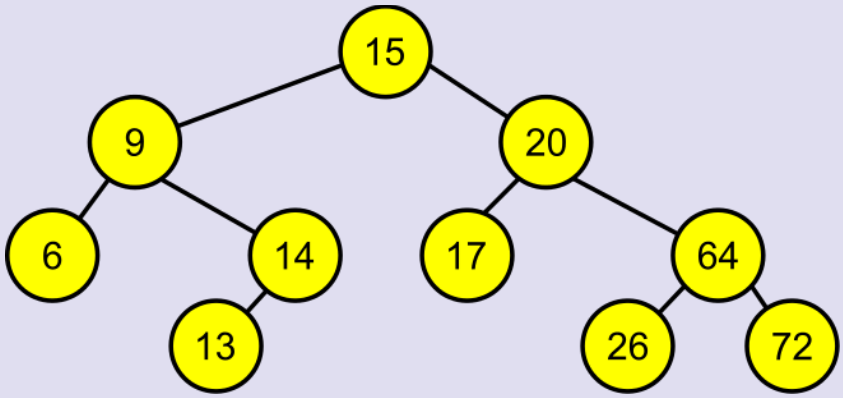
- **Ordenamiento por Inserción**: Tomo la primera y la coloco en mi mano. Luego tomo la segunda y la comparo con la que tengo: si es mayor, la pongo a la derecha, y si es menor a la izquierda. Después tomo la tercera y la comparo con las que tengo en la mano, desplazándola hasta que quede en su posición final. Continúo haciendo esto, insertando cada carta en la posición que le corresponde, hasta que las tengo todas en orden.



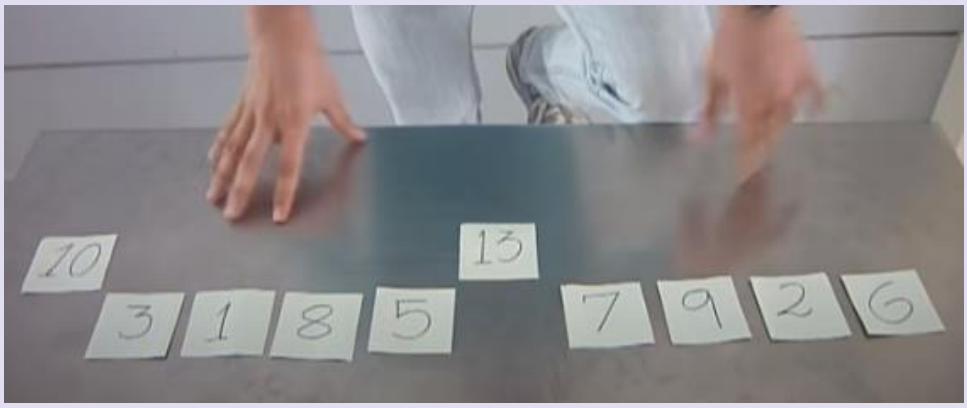
- **Ordenamiento por selección**: Buscas el elemento más pequeño de la lista. Lo intercambias con el elemento ubicado en la primera posición de la lista. Buscas el segundo elemento más pequeño de la lista. Lo intercambias con el elemento que ocupa la segunda posición en la lista. Repites este proceso hasta que hayas ordenado toda la lista.



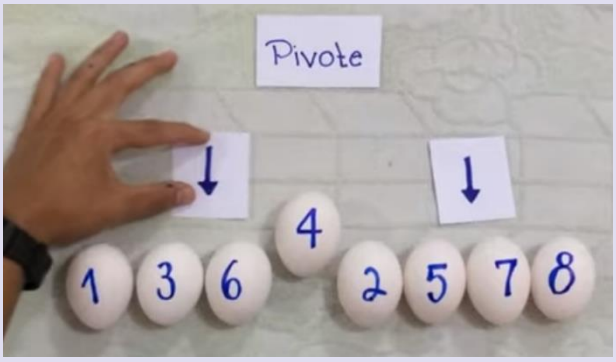
- **Ordenamiento con árbol binario**: ordena sus elementos haciendo uso de un árbol binario de búsqueda. Se basa en ir construyendo poco a poco el árbol binario introduciendo cada uno de los elementos, los cuales quedarán ya ordenados. Después, se obtiene la lista de los elementos ordenados recorriendo el árbol en *inorden*.



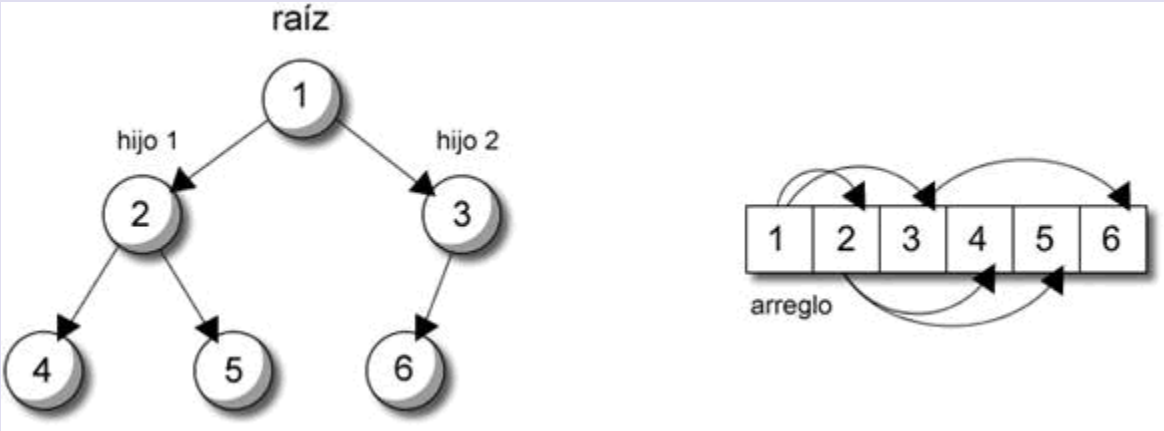
- **Ordenamiento Shell**: Consiste en dividir el arreglo (o la lista de elementos) en intervalos (o bloques) de varios elementos para organizarlos después por medio del ordenamiento de inserción directa.



- **Ordenamiento rápido (Quicksort)**: Es un algoritmo basado en la técnica de divide y vencerás



- **Ordenamiento por montículos (Heapsort)**: Consiste en almacenar todos los elementos del vector a ordenar en un montículo (*heap*), y luego extraer el nodo que queda como nodo raíz del montículo (cima) en sucesivas iteraciones obteniendo el conjunto ordenado. Basa su funcionamiento en una propiedad de los montículos, por la cual, la cima contiene siempre el menor elemento (o el mayor, según se haya definido el montículo) de todos los almacenados en él.



FIN DE CLASE-

POST CLASE-

Instalamos PSeint

A=3 ASIGNACIÓN

A==3 COMPARACIÓN

¡Necesito repasar todos los algoritmos de búsqueda! TODOS. Porque los entiendo pero sé que los olvidaré si no juego con ellos.