TRABAJO INDIVIDUAL: PREPARAR SANDWICH

1. Comprar un pan/ sacar el pan de la bolsa
2. Ocupar cuchillo para abrirlo
3. Poner mantequilla al pan
4. Agregar queso/ jamón de su preferencia
5. Cerrar el pan
6. Comerlo

TRABAJO GRUPAL

ALGORITMO: CAMBIAR PAÑAL A BEBE

Entrada: una guagua que necesita ser mudada de pañal, disponer a mano los materiales de aseo para bebé como agua tibia, toalla, toalla húmeda, pomada coceduras, polvo de féculas, mudador, ropa limpia, pañal limpio.

Proceso: acostar guagua en mudador, desvestir la guagua, quitar pañal sucio, arrojar en basurero, lavar guagua, secar, aplicar crema o polvo, volver a vestir a guagua.

Salida: Guagua ya mudada y cómoda

<https://javadesdecero.es/basico/tipos-datos-java-ejemplos/>

algoritmo Sumar

variables

entero num1, num2, suma

inicio

escribir "Introduzca primer número (entero): "

leer num1

escribir "Introduzca segundo número (entero): "

leer num2

suma = num1+num2

Escribir "El resultado de la suma es: ", suma

fin

ahi esta

2 ejercicio

Ejercicio 1

Determinar las variables y/o constantes involucradas

1)Realizar un algoritmo en pseudocodigo que permita sumar dos números.

variables: int n1, n2, suma

inicio

escribir "Introducir n1 (entero)

leer n1

escribir "Introducir n2 (entero)

leer n2

suma = n1+n2

Escribir "El resultado de la suma es: ", suma

fin

2) Realizar un algoritmo en Pseudocodigo que permita calcular el promedio de 3 notas

variables: nota1, nota2, nota3 promedio

inicio

escribir "Introducir primera nota (decimal)

leer nota1

escribir "Introducir segunda nota (decimal)

leer nota2

escribir "Introducir tercera nota

leer nota3 (decimal)

promedio = nota1+nota2+nota3/3

Escribir "El promedio de las 3 notas es: ", promedio

fin

3) Realizar un algoritmo en pseudocodigo que permita calcular el cuadrado de un n°

variables: n entero , resultado

inicio

escribir "introducir el valor de n entero"

leer n entero

cuadrado= n\*n

Escribir "el Cuadrado del número es" = resultado

LIQUIDACION SUELDO

Variables: rut, nombre, cargo

Haberes imponibles: SUELDO base, gratificación, hhee

Haberes no imponible: Asignación movilización, familiar, colación

INICIOnicio imprimir:"ingrese nombre" leer nombre imprimir:"ingrese rut" leer rut imprimir:"ingrese edad" leer edad imprimir:"ingrese cargo" leer cargo imprimir:"ingrese dias trabajados" leer diasTrabajados imprimir:"ingrese descuentos" leer descuentos imprimir:"ingrese sueldo diario" leer sueldoDiario sueldoMensual= (sueldo diario + dias trabajados )-descuentos imprimir: "su sueldo mensual es "+ sueldoMensual

nombre rut edad cargo díasTrabajados descuentos sueldoDiario sueldoMensual Inicio imprimir:"ingrese nombre" leer nombre imprimir:"ingrese rut" leer rut imprimir:"ingrese edad" leer edad imprimir:"ingrese cargo" leer cargo imprimir:"ingrese dias trabajados" leer diasTrabajados imprimir:"ingrese descuentos" leer descuentos imprimir:"ingrese sueldo diario" leer sueldoDiario sueldoMensual= (sueldo diario + dias trabajados )-descuentos imprimir: "su sueldo mensual es "+ sueldoMensua

LGORITMOS DE BÚSQUEDA

Búsqueda Secuencial

Búsqueda Binaria

Búsqueda Hashing

ALGORITMOS DE ORDENAMIENTO

Ordenamiento de Burbuja

Ordenamiento por Inserción

Ordenamiento por selección

Ordenamiento con árbol binario

Ordenamiento Shell

Ordenamiento rápido (Quicksort)

Ordenamiento por montículos (Heapsort)

Na

Natalia Carolina Gatica Gallardo

12:03

Nico no hay que resolver el ejercicio

se buscaba información

Na

Natalia Carolina Gatica Gallardo

12:05

https://jorgecontrerasp.wordpress.com/unidad-i/algoritmos-de-busqueda-y-ordenamiento/#:~:text=Consiste%20en%20ir%20comparando%20el,arreglo%20hasta%20que%20se%20encuentra.&text=Para%20utilizar%20este%20algoritmo%2C%20el,se%20deben%20repetir%20los%20elementos.&text=En%20cada%20iteraci%C3%B3n%20el%20arreglo%20se%20divide%20en%20dos.

Na

Natalia Carolina Gatica Gallardo

12:05

en ese link se enciuentra la información

Na

Natalia Carolina Gatica Gallardo

12:06

si francisco

Enviar mensaje a Chat público

nt i = 1

para (i =1; i<=5 ; i++)

si ( i < 5 )

escribir ( i ", ")

sino

escribir ("i" )

lectura:

"1 ,2 ,3 , 4, 5"

fin

ALGORITMOS DE BÚSQUEDA

Búsqueda Secuencial

La búsqueda secuencial busca un elemento de una lista utilizando un valor destino llamado clave.

En una búsqueda secuencial, los elementos de una lista o vector se exploran en secuencia

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.

Búsqueda Hashing

El método llamado por transformación de claves (hash), permite aumentar la velocidad de búsqueda sin necesidad de tener los elementos ordenados. Cuenta también con la ventaja de que el tiempo de búsqueda es prácticamente independiente del número de componentes del arreglo. Trabaja basándose en una función de transformación o función hash (H) que convierte una clave en una dirección (índice) dentro del arreglo

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.

for (i=1; i<TAM; i++)

for j=0 ; j<TAM – 1; j++)

if (lista[j] > lista[j+1])

temp = lista[j];

lista[j] = lista[j+1];

lista[j+1] = temp;

Ordenamiento por Inserción

es un algoritmo de fácil aplicación que permite el ordenamiento de una lista.

Su funcionamiento consiste en el recorrido por la lista seleccionando en cada iteración un valor como clave y compararlo con el resto insertándolo en el lugar correspondiente.

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

Consiste en encontrar el menor de todos los elementos del vector e intercambiarlo con el que está en la primera posición. Luego el segundo mas pequeño, y así sucesivamente hasta ordenarlo todo

Ordenamiento con árbol binario

Ordenamiento Shell

Cualquier algoritmo de ordenación que intercambia elementos adyacentes (como los algoritmos burbuja, selección o inserción) tiene un tiempo promedio de ejecución de orden cuadrático (n2). El método Shell mejora este tiempo comparando cada elemento con el que está a un cierto número de posiciones llamado salto, en lugar de compararlo con el el que está justo a su lado. Este salto es constante, y su valor inicial es N/2 (siendo N el número de elementos, y siendo división entera).

Se van dando pasadas con el mismo salto hasta que en una pasada no se intercambie ningún elemento de sitio. Entonces el salto se reduce a la mitad, y se vuelven a dar pasadas hasta que no se intercambie ningún elemento, y así sucesivamente hasta que el salto vale 1.

Ordenamiento rápido (Quicksort)

La idea del algoritmo es simple, se basa en la división en particiones de la lista a ordenar, por lo que se puede considerar que aplica la técnica divide y vencerás. El método es, posiblemente, el más pequeño de código, más rápido, más elegante, más interesante y eficiente de los algoritmos de ordenación conocidos.

Ordenamiento por montículos

(Heapsort)

Ordenación por mezcla

Este algoritmo consiste básicamente en dividir en partes iguales la lista de números y luego mezclarlos comparándolos, dejándolos ordenados.

Si se piensa en este algoritmo recursivamente, podemos imaginar que dividirá la lista hasta tener un elemento en cada lista, luego lo compara con el que está a su lado y según corresponda, lo sitúa donde corresponde.