BUAP®

VIRTUAL

BUAP

Metodología de la Programación

UNIDAD 2 Arreglos, cadenas y registros

Buscar y ordenar

Los métodos de búsqueda y ordenamiento pueden ser aplicados en diferentes áreas que requieren de la manipulación de una gran cantidad de datos.

Fuente
https://es.calameo.com/books/005170121a6a13e236ed6

Búsqueda

Proceso mediante el cual podemos localizar un elemento con un valor específico dentro de un conjunto de datos.

Fuente
https://biblioguias.uam.es/tutoriales/WOS/busqueda_basica

Valor a Buscar = 21

A[1]=1 <> VALOR

A = **1** 11 21 25 26 33 38 40 42 48

i=1

Segunda iteración: A[2]=11 <> VALOR

A = 1 **11** 21 25 26 33 38 40 42 48

i=2

Tercera iteración: A[3]=21 = VALOR

A = 1 11 **21** 25 26 33 38 40 42 48

i=3

Fuente https://myloview.es/cuadro-stickman-lapiz-lista-de-verificacion-no-23F2F04

Ejemplo


```
Busca_Sec()
  N    20
Inicio
Variables: A[N],i,b,band←0 : Entero
Para (i → 1 Hasta N, Incremento 1)
       Si(a[i]  =   b) entonces
          band  →  1
       Fin_Si
Fin_Para
    Si band = 1
        Escribir(“Encontrado”)
    Si_No
    Escribir (“Valor no encontrado”)
    Fin_Si
Fin_Busca_Sec
```

Búsqueda Binaria

- El método requiere que la información sobre la cual se va a buscar este ordenada.

- Al estar ésta ordenada puede descartarse la mitad que se sabe no es posible que este la información

Fuente https://www.cronista.com/clase/trendy/Marie-Kondo-de-oficina-10-consejos-para-ordenar-tu-escritorio-y-tus-ideas--20190417-0001.html

Algoritmo

```
Busca_Bin
N   20
Inicio
Variables: A[N],m,i,j,VALOR,POS: Entero
m → N / 2
   i → 1   j → N
   Coment: Recorrido del arreglo buscando VALOR
   Mientras ( ( A[ m ] <> VALOR ) AND ( i <= j ))
      Si A[m] > VALOR
         j → m – 1
      Si_No
         i → m + 1
       Fin_si
         m → ( i + j ) / 2
    Fin_Mientras
    Coment: Determinar si encontró o no
    Si (A[m] = VALOR)
    POS → m
Fin_Busca_Bin
```

A

VALOR=200

-3	1	6	100	120	150	200	346	678
1	2	3	4	5	6	7	8	9

Tipos de ordenamiento

- Ordenamiento interno.

  Se lleva a cabo completamente en memoria principal.

- Ordenamiento externo.

  No cabe toda la información en memoria principal y es necesario ocupar memoria secuencial

Fuente es.123rf.com/photo_7638324_hombres-con-cubos-números-3d-.html

Criterios de eficiencia

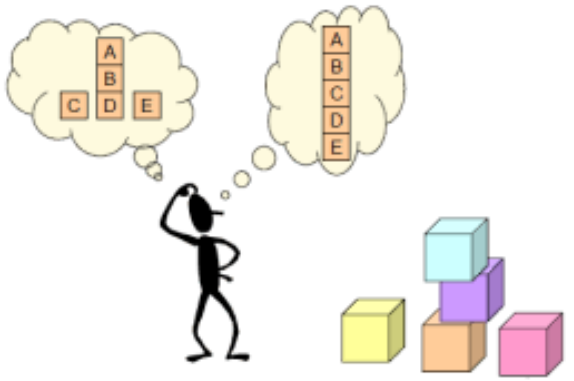
1. El número de pasos.
2. El número de comparaciones necesarios para ordenar n elementos.
3. El número de movimientos de elementos que se requieren para ordenar n elementos.

Fuente https://www.alamy.es/reloj-con-memoria-ram-tarjeta-forma-mascota-image248069289.html

Fuente https://www.freepik.es/vector-premium/ilustracion-dibujos-animados-reloj-sonriente-corriendo_6408477.htm

Métodos simples

- Método de Burbuja
- Método de Inserción
- Método de Selección



Fuente https://damardiazprogramacion.wordpress.com/acerca-de/

Fuente https://www.facebook.com/agentedeladiaria/

Burbuja

Caso medio

	1	2	3	4	5	6	7	8	
c	**25**	**3**	**12**	**19**	**2**	**1**	**9**	**6**	i
7	1	25	3	12	19	2	6	9	6
6		2	25	3	12	19	6	9	4
5			3	25	6	12	19	9	
4				6	25	9	12	19	
					9	25	12	19	
						12	25	19	
	1	2	3	6	9	12	19	25	

Mejor caso

1	2	3	6	9	12	19	25

```
Burbuja()
N  8
Inicio
  Variables: A[N]i,j,temp : Entero
  Para (i ←1 Hasta N, Incremento 1)
    Para (j ←N Hasta i+1 decremento 1)
      Si(a[j] < a[j-1])
        temp ← a[j]
        a[j]  ← a[j-1]
        a[j-1] ← temp
      Fin_si
    Fin_para
  Fin_para
Fin_burbuja
```

peor caso

25	19	12	9	6	3	2	1
1	2	3	6	9	12	19	25

Selección

25	3	12	19	2	1	9	6
1	3	12	19	2	25	9	6
1	2	12	19	3	25	9	6
1	2	3	19	12	25	9	6
1	2	3	6	12	25	9	19
				9	25	12	19
					12	25	19
						19	25

```
Selección()
N   8
Inicio
Variables: a[N],i,j,temp,imin:  Entero
  Para (i ← 1 Hasta N, Incremento 1)
    imin ← i
    Para (j ← i+1 Hasta N, Incremento 1)
      Si (a[j] < a [imin])
        imin ←  j
      Fin_si
    Fin_Para
    temp  ← a[i]
    a[i] ← a[imin]
    a[imin] ← temp
  Fin_Para
Fin_seleccion
```

Inserción

25	3	12	19	2	1	9	50
3	25	12					
3	12	25	19				
3	12	19	25	2			
2	3	12	19	25	1		
1	2	3	12	19	25	9	
1	2	3	9	12	19	25	50
1	2	3	6	9	12	19	25

```
Nombre: insercion
N  8
Entrada: a[N] ,i,j,temp : Entero
Inicio
 Para (i ← 2 Hasta N, Incremento 1)
    j ← i
   Mientras (j>1) AND (a[j] < a[j-1] )
      temp← a[i]
      a[i] ← a[j-1]
      a[j-1] ← temp
       j ← j-1
   Fin_Mientras
 Fin_Para
Fin_insercion
```

Inserción

```
Nombre: temperaturas
D   31
Entrada: enero[]
Inicio
   Variables: dia de tipo Entero
                    enero[D], Suma←0, Tprom : Entero
    Escribir("Dame 31 temperaturas")
    Para (dia←1 hasta 31 Paso 1)      coment: es recorrido del arreglo
        Leer(enero[dia])
    FinPara
    Para (dia←1 hasta 31 Paso 1)
         Suma←Suma+enero[dia]
    FinPara
    Tprom← Suma/D
     Escribir("en promedio tuvimos", Tprom, "grados de temperatura")
 Fintemperaturas
```

	enero
1	5
2	8
.	6
.	0
31	1

Inserción

n!=n*(n-1)*(n-2)* …*1
Sin arreglo

```
Nombre: Facto
Inicio
   Variables: n,term,fact←1 de tipo Entero
   Escribir("Dame un número")
   Leer(n)
   Para (term←n Hasta 1 paso -1)
      fact← fact * term
   Fin Para
FinFacto
```

Ejemplo si n=4
4!= 4*(4-1)*(4-2)*(4-3)
4*3*2*1

n	term	fac
4	4	4
	3	12
	2	24
	1	24

n!=n*(n-1)*(n-2)* …*1
Con arreglo
Nombre:Facto1

```
Inicio
   Variables: pos, pasa term[40],fac←1 de tipo
Entero
   Escribir("Dame un número")
   Leer(n)
   Para (pos←1 Hasta n Paso 1)
      term[pos]←n
      n←n-1
   FinPara
   Para (pasa←1 Hasta n Paso 1)
      fac← fac*term[pasa]
   FinPara
FinFacto1
```

	term
1	6
2	5
.	
.	
6	1

Bibliografía

1. Cairó O (2005). Metodología de la programación, Algoritmos, diagramas de flujo y programas (3ª ed). México: Alfaomega.
2. Joyanes, A. (2008). Fundamentos de programación, Algoritmos, Estructuras de datos y Objetos (4ª. ed). España: Mc Graw Hill.
3. Skiena, S (2008). The algorithm design Manual (2nd ed). USA: Springer.

Responsables del Curso

De la Rosa Flores Rafael

Moyao Martínez Yolanda

Sánchez Román Guillermina

gracias.

BUAP ©2020