

Sistemas de Datos

Práctico N° 1

Para cada una de las problemáticas planteadas se pide:

- Indicar que hace el algoritmo,
- computar el tiempo de ejecución y
- determinar el orden de complejidad.

a)

```
void AAAA (tipo &xp)
{ tipo p, nuevo, anterior;
  nuevo =(Tipo) malloc(sizeof(struct nodo));
  printf("\n Ingrese el nuevo valor: ");
  scanf("%d",&nuevo->nro);
  nuevo->sig = NULL;
  if (xp == NULL) /* */
    xp = nuevo;
  else
  { p = xp; /* */
    while (p != NULL)
    { anterior = p;
      p = p->sig;
    }
    anterior->sig = nuevo;
    printf("listo.");
  };
  return;
}
```

b)

```
void BBB(tipo &xp, int xnum)
{ puntero p, nuevo, anterior;
  nuevo =(Tipo) malloc(sizeof(struct nodo));
  nuevo->nro=xnum;
  if (xp == NULL) /* */
  { xp = nuevo;
    nuevo->sig =NULL;}
  else
  if (xp->nro == xnum) /* */
  { nuevo->sig = xp;
    xp=nuevo; }
  else
  { xp = xp->sig; /* */
    anterior=xp; /* */
    while ( (p != NULL) && (xnum > p->nro))
    { anterior = p;
      p = p->sig; }
    anterior->sig = nuevo;
    nuevo->sig = p;
    printf("\n Listo"); }
}
```

c)

```
#include <stdio.h>
#define n 11
int main()
{int cant[n],inferior=0,superior=n-1, m, i, num;
```

```

for(i=0; i < n; i++)
{ printf("Ingrese matrícula \n");
  scanf ("%d",&cant[i]); }
printf(" \n Ingrese una matricula ");
scanf("%d", &num);
m=(inferior + superior) / 2;
while ((inferior <= superior) && (cant[m] != num))
{ if (num < cant[m])
  superior=m - 1;
  else
  inferior= m + 1;
  m=(inferior + superior) / 2; }
if (inferior > superior)
  printf("\ el alumno no aprobó");
else
  printf("\ aprobó y está en %d ", m);
getchar();
}

```

d)

```

#include <stdio.h>
#define N 8
int main(void)
{ int t[N];
  int i, k,a,cota;
  printf("\n ingrese %d valores \n ");
  for( i=0; i < N; i++)
    scanf("%d",&t[i]);
  cota=N - 1;
  k=1;
  while( k!= -1)
  { k=-1;
    for (i=0; i < cota ; i++)
      if (t[i] > t[i+1])
      { a=t[i];
        t[i]=t[i+1];
        t[i + 1]=a;
        k=i; }
    cota=k; }
  printf("\n listo ");
  for (i=0; i < N; i++)
    printf("\n %d",t[i]);
  getchar();
}

```