1.

```
void algoritmo1(int n){
    int i, j = 1;                       //1
    for(i = n * n; i > 0; i = i / 2){   //log2(n^2)+2
        int suma = i + j;               //log2(n^2)+1
        printf("Suma %d\n", suma);      //^^
        ++j;                            //^^
    }
}
//1+log2(n^2)+2+log2(n^2)+1+log2(n^2)+1+log2(n^2)+1
//4log2(n^2)+6
//O(log(n))
```

Al ejecutar algoritmo1(8), obtenemos los valores: 65, 34, 19, 12, 9, 8, 8.

Este algoritmo eleva la entrada a la 2 y lo divide entre 2 hasta que llegue a 0, además el aalgoritmo imprime la variable suma la cual es la suma entre el valor de i en el loop actual y j la cual aumenta cada loop empezando desde uno.

2.

```
int algoritmo2(int n){
        int res = 1, i, j;                  //1
        for(i = 1; i <= 2 * n; i += 4){     //((n+1)/2)+1
            for(j = 1; j * j <= n; j++){    //(((n+1)/2)*sqrt(n))+1
                res += 2;                   //((n+1)/2)*sqrt(n)
            }
        }
        return res;                         //1
}

//1+((n+1)/2)+1+(((n+1)/2)*sqrt(n))+1+((n+1)/2)*sqrt(n)
//2((n+1)/2)*sqrt(n)+((n+1)/2)+3
//O(n*sqrt(n))
```

Al ejecutar algoritmo2(8), obtenemos el valor: 17.

3.

```
void algoritmo3(int n){
    int i, j, k;                            //1
    for(i = n; i > 1; i--){                 //n
        for(j = 1; j <= n; j++){            //(n^2)
            for(k = 1; k <= i; k++){        //(n^3)
                printf("Vida cruel!!\n");   //(n^3)-1
            }
        }
    }
}

//2(n^3)+(n^2)+n
//O(n^3)
```

4.

```
int algoritmo4(int* valores, int n){
    int suma = 0, contador = 0;             //1
    int i, j, h, flag;                      //1
    for(i = 0; i < n; i++){                 //n+1
        j = i + 1;                          //n
        flag = 0;                           //n
        while(j < n && flag == 0){          //worst case: (n^2)+1   best case:
n+2
            if(valores[i] < valores[j]){    //n
                for(h = j; h < n; h++){     //worst case: (n^3)+1   best case: 0
                    suma += valores[i];     //worst case: (n^3)     best case: 0
                }
            }else{                          //worst case: 0         best case: n
                contador++;                 //worst case: 0         best case: n
                flag = 1;                   //worst case: 0         best case: n
            }
            ++j;                            //worst case: (n^2)     best case: n
        }
    }
    return contador;                        //1
}

//best case: 8n+6
//O(n)

//worst case: 2((n^3)+(n^2)+2n+3)
//O(n^3)
```

Este algoritmo da como resultado el numero de elementos de una lista que son mayores a todos los números que le siguen.

5.

```c
void algoritmo5(int n){
    int i = 0;                              //1
    while(i <= n){                          //7
        printf("%d\n", i);                  //6
        i += n / 5;                         //6
    }
}

//20
//O(1)
```

6.

```python
def fibo(n):
    if n==0:                            #case n=0: 1        case n=1: 1         case n>1: 1
        return 0                        #case n=0: 1        case n=1: 0         case n>1: 0
    elif n==1:                          #case n=0: 0        case n=1: 1         case n>1: 0
        return 1                        #case n=0: 0        case n=1: 1         case n>1: 0

    return fibo(n-1)+fibo(n-2)          #case n>1: n^2

#case n>1: (n^2)+1
#O(n^2)

#case n=0: 2
#O(1)

#case n=1: 3
#O(1)
```

Tamaño de entrada	Tiempo	Tamaño de entrada	Tiempo
5	26	35	1226
10	101	40	1601
15	226	45	2026
20	401	50	2501
25	626	60	3601
30	901	100	10001

7.

```
def fiboLoop(n):
    if n==0:                            #case n=0: 1        case n=1: 1         case n>1: 1
        return 0                        #case n=0: 1        case n=1: 0         case n>1: 0
    elif n==1:                          #case n=0: 0        case n=1: 1         case n>1: 0
        return 1                        #case n=0: 0        case n=1: 1         case n>1: 0

    a,b=0,1                             #case n>1: 1

    for i in range(2, n+1):             #case n>1: n+1
        c=a+b                           #n
        a,b=b,c                         #n

    return b                            #1

#case n>1: 3n+4
#O(n^2)

#case n=0: 2
#O(1)

#case n=1: 3
#O(1)
```

Tamaño de entrada	Tiempo	Tamaño de entrada	Tiempo
5	19	45	139
10	34	50	154
15	49	100	304
20	64	200	604
25	79	500	1504
30	94	1000	1004
35	109	5000	15004
40	124	10000	30004