1.- Determinar la complejidad asintótica $O(f(n))$ del siguiente pseudo código y obtener su función de tiempo $T(n)$ aproximada

Funcion algoritmo $S(n)$

```
cont ← 2^n           --->1
for j←1 to n do      --->n
    s ← cont         --->1
    while s≥1 do     --->K
        s←s/2        --->2
    end while
end for
return S
```

$1+(n)(1+2(K))$

$1+(n)(1+2n)$

$1+n+2n^2$

$2n^2+n+1 = T(n)$

$O(n^2)$

$S_0=S$ Condición

$S_1=S/2^1$ $S_K=1$

$S_2=S/2^2$

$S_K=S/2^K$

$1=S/2^K$ $2^K=S$

$2^K=2^n$ $\boxed{K=n}$

2.- Utilizando el método de iteración obtener la complejidad asintótica $O(f(n))$ de la siguiente recurrencia

$T(n)=\underline{2T(n/2)+n}$; $T(1)=1$

Asumir: $n=2^K$

$\log n=K$

$= 2(2T(n/4)+\frac{n}{2})+n$

$= 4T(n/4)+n+n$

$= \underline{4T(n/4)+2n}$

$= 4(2T(n/8)+\frac{n}{4})+2n$

$= \underline{8T(n/8)+3n}$

$= 2^K T(n/2^K)+Kn$

$= nT(1)+n\log n$

$\underline{O(n\log n)}$ //