



Esercizio 0

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es0(n):  
    t=0  
    n=abs(n)  
    if n>100: return 1  
    for i in range(n):  
        t+=3  
    return t
```

Esercizio 1

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es1(n):  
    if n<0: n=-n  
    while n:  
        if n%2: return 1  
        n-=2  
    return 0
```

Esercizio 2

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es2(n):  
    n=abs(n)  
    x=r=0  
    while x*x<n:  
        x+=1  
        r+=3*x  
    return r
```

Esercizio 3

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es3(n):  
    n=abs(n)  
    x=r=0  
    while n>1:  
        r+=2  
        n=n//3  
    return r
```

Esercizio 4

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es4(n):  
    n=abs(n)  
    t=x=1  
    for i in range(n):  
        t=3*t  
    while t>=x:  
        x+=2  
        t-=x  
    return x
```

Esercizio 5

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es5(n):  
    n=abs(n)  
    p=2  
    while n >=p:  
        p=p*p  
    return p
```

Esercizio 6

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es6(n):  
    n=abs(n)  
    i,j,t,s=1  
    while i<=n:  
        for j in range(t):  
            s+=1  
        i=i*i  
        t+=1  
    return s
```

Esercizio 7

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es7(n):
    n=abs(n)
    s=t=n
    p=0
    while s>=1:
        s=s//4
        p+=1
    while n-s>0:
        n-=s
        t+=5
    return t
```

Esercizio 8

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es8(n):
    n=abs(n)
    s=n
    p=2
    i, r=1
    while s>=1:
        s=s//5
        p+=2
    p=p*p
    while i*i*i<n:
        for j in range(p):
            r+=1
        i+=1
    return r
```

Esercizio 9

Calcolare il costo del seguente algoritmo, distinguendo tra caso migliore e caso peggiore se necessario.

```
def es9(n):
    tot=n
    if n<=100: return 5
    j=512
    while j>=2:
        k=1
        while k*k<=n:
            k=2*k
        tot=tot+5
        j=j//2
    return tot
```

Soluzioni

- Esercizio 0: $\Theta(1)$
- Esercizio 1:
 - caso migliore $\Theta(1)$ (quando n è dispari)
 - caso peggiore $\Theta(n)$ (quando n è pari)
- Esercizio 2: $\Theta(\sqrt{n})$
- Esercizio 3: $\Theta(\log n)$
- Esercizio 4: $\Theta(3^n)$
- Esercizio 5: $\Theta(\log \log n)$
- Esercizio 6: $\Theta(n)$
- Esercizio 7: $\Theta\left(\frac{n}{\log n}\right)$
- Esercizio 8: $\Theta(\sqrt[3]{n} \log^2 n)$
- Esercizio 9: $\Theta(\log n)$