

Costo computazionale: esercizi da risolvere

Tiziana Calamoneri



SAPIENZA
UNIVERSITÀ DI ROMA

Esercizi per casa

Calcolare il costo del seguente algoritmo scritto in pseudocodice, distinguendo tra caso migliore e caso peggiore se necessario:

```
def Selection_Sort(A):  
    for i in range(len(A)-1):  
        m=i  
        for j in range(i+1,len(A)):  
            if A[j] < A[m]:  
                m = j  
        A[m],A[i]=A[i],A[m]
```

Esercizi per casa

Calcolare il costo del seguente algoritmo scritto in pseudocodice, distinguendo tra caso migliore e caso peggiore se necessario:

```
def Bubble_Sort(A)
    for i in range(len(A)-1):
        for j in range(len[A]-i-1):
            if (A[j] > A[j+1]):
                A[j],A[j+1]=A[j+1],A[j]
```

Analogo al precedente.
Fare per esercizio se non già risolto...

Esercizi per casa

Calcolare il costo del seguente algoritmo scritto in pseudocodice, distinguendo tra caso migliore e caso peggiore se necessario:

```
def Insertion_Sort(A):
    for j in range (1,len(A)):
        x = A[j]
        i = j - 1
        while (i>=0) and (A[i]>x):
            A[i+1] = A[i]
            i=i-1
        A[i+1]=x
```

caso migliore e peggiore qui sono diversi...

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)  
    x=t=1  
    for i in range(n):  
        t=3*t  
    while t>=x:  
        x+=2  
        t-=2  
    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*i<=n:  
        for j in range(t)  
            s+=1  
        i=i+1  
        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)  
    t,s=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)  
    t,s=n  
    p=0  
    while s>=1:  
        s=s//4  
        p+=1  
    while n-p>0:  
        n-=p  
        t+=5  
    return t
```

Esercizio 9

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

```
def es9(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 10

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

```
def es10(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:
Caso migliore=peggiore: $\Theta(1)$

Esercizio 1:
Caso migliore: $\Theta(1)$ (quando n è dispari)
Caso peggiore: $\Theta(n)$ (quando n è pari)

Esercizio 2:
Caso migliore=peggiore: $\Theta(n^{1/2})$

Esercizio 3:
Caso migliore=peggiore: $\Theta(\log n)$

Esercizio 4:
Caso migliore=peggiore: $O(3^n)$

Esercizio 5:
Caso migliore=peggiore: $\Theta(\log \log n)$

Esercizio 6:
Caso migliore=peggiore: $\Theta(n)$

Esercizio 7:
Caso migliore=peggiore: $\Theta(n)$

Esercizio 8:
Caso migliore=peggiore: $\Theta(n / \log n)$

Esercizio 9:
Caso migliore=peggiore: $\Theta(n^{1/3} \log^2 n)$

Esercizio 10:
Caso migliore=peggiore: $\Theta(\log n)$