**Es 1.**
**Quali delle seguenti affermazioni sono vere e quali false?**
**<u>Si motivi la risposta caso per caso.</u>**

1. $n^2 = O(2^n)$
2. $n^2 = \Theta(2^n)$
3. $8^n = O(4^n)$
4. $8^n = \Omega(4^n)$

**Soluzione**

**1.**
**$n^2 = O(2^n)$ è vera perchè ogni funzione esponenziale cresce più in fretta di un polinomio.**
**Si deve dimostrare che esistono due costanti c, $n_0 \geq 0$ tali che $0 \leq n^2 \leq c\ 2^n$ per ogni $n \geq n_0$.**
**Possiamo ridurci a una disuguaglianza più semplice applicando il logaritmo a entrambi i membri.**
**Quindi**
**$n^2 \leq c\ 2^n, \Leftrightarrow \lg n^2 \leq \lg c\ 2^n \Leftrightarrow 2 \lg n \leq \lg c + n \leq (\lg c)\ n$ che è vera per $c = 4$ e per ogni $n \geq n_0 = 1$.**

**Es 1.**
**Quali delle seguenti affermazioni sono vere e quali false?**
**<u>Si motivi la risposta caso per caso.</u>**

2. $n^2 = \Theta(2^n)$
3. $8^n = O(4^n)$
4. $8^n = \Omega(4^n)$

**Soluzione**

**2.**
**$n^2 = \Theta(2^n)$ è falsa perchè una funzione esponenziale non può limitare inferiormente una funzione polinomiale.**

**Possiamo confutare l'esistenza di due constanti c ed $n_0$ tali che $n^2 \geq c2^n$, per ogni $n \geq n_0$ , facendo vedere per ogni scelta di c esiste un n per cui è falsa.**

**Per esempio prendendo n = c allora dovrebbe essere $c^2 \geq c2^c \Leftrightarrow c \geq 2^c$ e questo è falso per ogni c≥0**

**Es 1.**
**Quali delle seguenti affermazioni sono vere e quali false?**
**<u>Si motivi la risposta caso per caso.</u>**

3. $8^n = O(4^n)$
4. $8^n = \Omega(4^n)$

**Soluzione**

**3.**
**$8^n = O(4^n)$ è falso perchè possiamo confutare l'esistenza di due constanti c ed $n_0$ tali che $8^n = \leq c4^n$, facendo vedere per ogni scelta di c è falsa.**
**Dovrebbe essere $8^n \leq c4^n \Leftrightarrow 4^n * 2^n \leq c4^n$, perchè $8 = 2*4$ e quindi $8^n = 4^n * 2^n$, e cioè che $2^n \leq c$, ma nessuna costante può limitare superiormente una funzione crescente, basta prendere n=c e si ottiene di nuovo $c \geq 2^c$ che è falso per ogni $c \geq 0$.**

**Es 1.**
**Quali delle seguenti affermazioni sono vere e quali false?**
**<u>Si motivi la risposta caso per caso.</u>**

**4.** $8^n = \Omega(4^n)$

**Soluzione**

**4.**
**$8^n = \Omega(4^n)$ è vera, perchè $8^n$ cresce più in fretta di $4^n$.**
**In dettaglio $8^n \geq c4^n \Leftrightarrow 4^n * 2^n \geq c4^n \Leftrightarrow 2^n \geq c$ e questo è vero per ogni $n \geq n_0 = c$.**

**Es 2. Si analizzi la seguente funzione esprimendo il tempo di esecuzione asintotico nei casi peggiore e migliore. In questa soluzione assumiamo che l'input n sia non negativo.**

```
analizza(n)
{
  count = 0;
 if n è dispari then i = n else i = 2^n
 while i > 0
        for (j = 0; j < i; j++)
              count += 1;
        i= i/2
return count;
}
```

**Sol. Il ciclo for interno è eseguito i volte, dobbiamo sommare tutti i valori di i per sapere quante volte viene eseguito essendo inserito in un ciclo while.**

**Il caso peggiore si verifica quando n è pari, infatti in questo caso $i = 2^n$**
**e quindi il ciclo for viene eseguito**
**$2^n$ volte all'inizio quando $i = 2^n$**
**$2^{n-1}$ volte la seconda volta quando $i = 2^{n-1}$**

**…**
**1 volta l'ultima volta quando $i = 2^0$**

**Sommando otteniamo un totale di $2^{n+1}$ -1 volte. Quindi possiamo dire che nel caso peggiore la funzione *analizza* ha un tempo di esecuzione in $\Theta(2^n)$**

**Es 2. Si analizzi la seguente funzione esprimendo il tempo di esecuzione asintotico nei casi peggiore e migliore. In questa soluzione assumiamo che l'input n sia non negativo.**

```
analizza(n)
{
  count = 0;
 if n è dispari then i = n else i = 2^n
 while i > 0
        for (j = 0; j < i; j++)
              count += 1;
        i= i/2
return count;
}
```

**Sol. Il ciclo for interno è eseguito i volte, dobbiamo sommare tutti i valori di i per sapere quante volte viene eseguito essendo inserito in un ciclo while.**

**Nel caso migliore invece i=n e quindi le istruzioni del ciclo for vengono eseguite**
**n volte la prima volta quando i = n**
**n/2 volte la seconda volta quando i = n/2**
**…**
**1 volta l'ultima volta quando $i = n/2^k$ con $2^k \leq n < 2^{k+1}$ per cui $2^k /2^k \leq n/2^k < 2^{k+1} /2^k = 2$**

**per un totale di $n\Sigma_{i=o\ldots k}(1/2)^i$ volte. Poiché possiamo maggiorare la sommatoria con la sua estensione all'infinito che converge a una costante, possiamo dire che nel caso migliore la funzione *analizza* ha un tempo di esecuzione in O(n).**
**Osservando che il ciclo for è eseguito almeno n volte, otteniamo che il tempo di esecuzione è anche in Ω(n) e in conclusione in Θ(n)**

**Es. 3 Supponiamo che un algoritmo abbia un tempo di esecuzione nel caso peggiore espresso da f(n) e quello del caso migliore espresso da g(n).**
**Quali tra queste affermazioni sono vere?**
**<u>Si motivi la risposta</u>**

**1. f(n) = O(g(n))**
**2. f(n) = Ω(g(n))**
**3. g(n) = O(f(n))**
**4. g(n) = Ω(f(n))**

**Sol.**
**1. f(n) = O(g(n)) è falsa in generale, sarà vera solo se il tempo di esecuzione nei due casi è lo stesso. Infatti Il tempo di esecuzione nel caso migliore, g(n), fornisce un limite inferiore al tempo di esecuzione in tutti i casi, non superiore.**

**2. f(n) = Ω(g(n)) è sempre vera, perchè g(n) è il tempo di esecuzione nel caso migliore che fornisce un limite inferiore al tempo di esecuzione in tutti i casi.**

**3. Il tempo di esecuzione nel caso peggiore fornisce un limite superiore al tempo di esecuzione in tutti i casi, quindi g(n) = O(f(n)) è certamente sempre vera**

**4. g(n) = Ω(f(n)) in generale è falsa, perchè il tempo di esecuzione nel caso peggiore fornisce un limite superiore al tempo di esecuzione in tutti i casi; è vera solo quando i tempi di esecuzione nei due casi coincidono.**