**Algoritmi e Strutture Dati** **Anno Accademico 2022–2023**

# Esercizi: Notazione Asintotica e Costo Computazionale

*Manuela Montangero*

1. *Esercizio.* Per ognuna delle seguenti funzioni determinare la notazione $\Theta$ appropriata.

   (a) $6n + 1$

   (b) $6n^3 + 12n^2 + 1$

   (c) $2 \log n + 4n + 3n \log n$

   (d) $3n^2 + 2n \log n$

   (e) $(6n + 1)^2$

   (f) $1 + 2 + 4 + 8 + 16 + \ldots + 2^n$

   *Soluzioni:* Gli esercizi si possono risolvere utilizzando (1) le regole indicate nel file *Notazione asintotica* e/o (2) la definizione delle notazioni asintotiche.

   (a) (1) $6n + 1$ è un polinomio di primo grado, quindi $\in \Theta(n)$.
   (2) Abbiamo che $6n + 1 \geq 6n$ per ogni $n \geq 1$, quindi $6n + 1 \in \Omega(n)$ ($c = 6$ e $n_0 = 1$). Analogamente $6n + 1 \leq 7n$ per ogni $n \geq 1$, quindi $6n + 1 \in O(n)$ ($c = 7$ e $n_0 = 1$). Se ne conclude che $6n + 1 \in \Theta(n)$.

   (b) (1) $6n^3 + 12n^2 + 1$ è un polinomio di terzo grado, quindi $\in \Theta(n^3)$.
   (2) Abbiamo che $6n^3 + 12n^2 + 1 \geq 6n^3$ per ogni $n \geq 0$, quindi $6n^3 + 12n^2 + 1 \in \Omega(n^3)$ ($c = 6$ e $n_0 = 0$). Inoltre,
   $$6n^3 + 12n^2 + 1 \leq 6n^3 + 12n^3 + n^3 = 19n^3 \text{ per } n > 0,$$
   quindi $6n^3 + 12n^2 + 1 \in O(n^3)$ ($c = 19$ e $n_0 = 1$). Se ne conclude che $6n^3 + 12n^2 + 1 \in \Theta(n^3)$.

   (c) (2) Abbiamo che $2 \log n + 4n + 3n \log n \geq 3n \log n$, quindi $2 \log n + 4n + 3n \log n \in \Omega(n \log n)$ ($c = 3$ e $n_0 = 1$). Inoltre,
   $$2 \log n + 4n + 3n \log n \leq 2n \log n + 4n \log n + 3n \log n = 9n \log n \in O(n \log n).$$
   Quindi, $2 \log n + 4n + 3n \log n \in \Theta(n \log n)$.

   (d) (2) Abbiamo che $3n^2 + 2n \log n \geq 3n^2 \in \Omega(n^2)$ ($c = 3$ e $n_0 = 1$). Inoltre, $3n^2 + 2n \log n \leq 3n^2 + 2n^2 = 5n^2 \in O(n^2)$ ($c = 5$ e $n_0 = 1$). Per cui $3n^2 + 2n \log n \in O(n^2)$.

   (e) (1) La funzione è un polinomio di secondo grado: $(6n + 1)^2 = 36n^2 + 12n + 1 \in \Theta(n^2)$.

   (f) (2) $1 + 2 + 4 + 8 + 16 + \ldots + 2^n = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \in \Theta(2^n)$ perchè $2^n \leq 2^{n+1} - 1 \leq 4 \cdot 2^n$, per $n \geq 1$.

2. *Esercizio.* Sia $f(n) = n + 2n^3 - 3n^3 + 4n^4$. Dire quali delle seguenti affermazioni sono vere e quali false. Giustificare le risposte.

   (a) $f(n) \in \Omega(n \log n)$

   (b) $f(n) \in \Theta(n^5)$

   (c) $f(n) \in O(n^{10})$

   (d) $f(n) \in \Omega(n^4)$

*Soluzione.* Abbiamo $f(n) = n + 2n^3 - 3n^3 + 4n^4 = 4n^4 - n^3 + n \in \Theta(n^4)$. Per cui (in base alle tabelle nel file *Notazione asintotica*): (a) Vero; (b) Falso; (c) Vero; (d) Vero.

3. *Esercizio.* Sia $f(n) = n\log n + 2n^3 - 3n^2$. Dire quali delle seguenti affermazioni sono vere e quali false. Provare le affermazioni che si ritengono vere.

   (a) $f(n) \in \Omega(n \log n)$

   (b) $f(n) \in \Theta(n^3)$

   (c) $f(n) \in O(n^4)$

   (d) $f(n) \in \Omega(n^2)$

   (e) $f(n) \in \Omega(n^5)$

   *Soluzione.* Abbiamo

   $$f(n) = 2n^3 - 3n^2 + n\log n \leq 2n^3 + n\log n \leq 2n^3 + n^3 = 3n^3 \text{ per } n > 0,$$

   quindi $f(n) \in O(n^3)$. Inoltre, $2n^3 - 3n^2 + n\log n \geq 2n^3 - 3n^2$. Per provare che $f(n) \in \Omega(n^3)$ dobbiamo far vedere che esiste una costante $c > 0$ e un $n_0 \geq 0$ per cui vale

   $$2n^3 - 3n^2 \geq c \cdot n^3, \quad \forall n \geq n_0.$$

   Cerchiamo queste due costanti tali che:

   $$\begin{array}{rcl} 2n^3 - 3n^2 & \geq & c \cdot n^3 \text{ dividiamo entrambi i lati per } n^2 > 0 \text{ (se } n > 0) \\ 2n - 3 & \geq & cn \\ 2n - cn & \geq & 3 \\ (2-c)n & \geq & 3 \text{ dividiamo entrambi i lati per } 2 - c > 0 \text{ se } c < 2 \\ n & \geq & \dfrac{3}{2-c} \end{array}$$

   Quindi, se fissiamo $c = 1 < 2$, per $n \geq 3$, abbiamo che $2n^3 - 3n^2 \geq n^3 \in \Omega(n^3)$.

   Per cui (in base alle tabelle nel file *Notazione asintotica*): (a) Vero; (b) Vero; (c) Vero; (d) Vero; (e) Falso.

   **IN GENERALE** possiamo usare i seguenti risultati (per evitare di dover fare tanti conti ogni volta):

   - se $f(n) \in O(g(n))$ e $d(n) \in O(h(n))$, allora $f(n) + g(n) \in O\left(\max\{g(n), h(n)\}\right)$.

     Intuitivamente, la somma è dominata dal termine di ordine più grande.

     Quindi, nell'esercizio 1.d, scegliendo $f(n) = 3n^2 \in O(n^2)$ e $d(n) = 2n\log n \in O(n\log n)$, possiamo concludere che $f(n) + h(n) = 3n^2 + 2n\log n \in O(n^2)$ perchè $\max\{g(n), h(n)\} = n^2$.

   - se $f(n) \in \Omega(g(n))$, $d(n) \in O(h(n))$ e $h(n)$ viene prima di $g(n)$ nella tabella delle $O(\cdot)$ (sempre quella del file *Notazione asintotica*), allora $f(n) - g(n) \in \Omega\left(g(n)\right)$.

     Intuitivamente, nella differenza, se il sottraendo ha ordine di grandezza minore dell'ordine di grandezza del minuendo, domina il minuendo.

     Nell'esercizio 3, abbiamo $f(n) \in O(n^3)$. Ripartendo da $2n^3 - 3n^2$ abbiamo che $2n^3 \in \Omega(n^3)$ e $3n^2 \in O(n^2)$. Quindi, $f(n) \in \Omega(n^3)$.

4. *Esercizio.* Selezionare la notazione $\Theta$ opportuna tra $\Theta(1), \Theta(\log n), \Theta(n), \Theta(n \log n), \Theta(n^2), \Theta(n^3), \Theta(2^n), \Theta(n!)$ per indicare il numero di volte in cui l'istruzione $x := x + 1$ viene eseguita.

   (a) for $i = 1$ to $2n$
       $x := x + 1$

   *Soluzione.* Il numero di volte in cui l'istruzione $x := x + 1$ viene eseguita è uguale al numero di volte in cui viene eseguito il ciclo `for`, che possiamo scrivere così:

   $$\sum_{i=1}^{2n} 1 = 2n \in \Theta(n),$$

   dove abbiamo usato la formula per calcolare la somma dei primi $k$ interi. In questo caso $k = 2n$.

   (b) for $i = 1$ to $2n$
       for $j = 1$ to $n$
         $x := x + 1$

   *Soluzione.* Ogni volta che il ciclo interno viene eseguito, l'istruzione viene eseguita $\sum_{j=1}^{n} 1 = n$ volte. Il ciclo interno viene eseguito ad ogni iterazione del ciclo esterno, cioè $2n$ volte. Quindi, in totale l'istruzione $x := x + 1$ viene eseguita

   $$2n \cdot n \in \Theta(n^2)$$

   volte.

   (c) for $i = 1$ to $2n$
       for $j = 1$ to $i$
         for $k = 1$ to $i$
           $x := x + 1$

   *Soluzione.* Il ciclo più interno esegue l'istruzione $x := x + 1$ una volta ad ogni iterazione, ovvero per un totale di $\sum_{k=1}^{i} 1 = i$. Il ciclo più interno viene eseguito ad ogni iterazione del secondo ciclo (quello con indice $j$), che a sua volta viene eseguito una volta ad ogni iterazione del ciclo più esterno. Quindi, mettendo tutto insieme, abbiamo (guardate bene gli indici delle sommatorie e quelle del codice):

   $$\begin{aligned}
   \underbrace{\sum_{i=1}^{2n}}_{\text{ciclo esterno}} \underbrace{\sum_{j=1}^{i}}_{\text{secondo ciclo}} \overbrace{\underbrace{\sum_{k=1}^{i} 1}_{\text{ciclo interno}}}^{=i} &= \sum_{i=1}^{2n} \underbrace{\sum_{j=1}^{i} i}_{(*)} \\
   &= \sum_{i=1}^{2n} i \overbrace{\underbrace{\sum_{j=1}^{i} 1}}^{=i} \\
   &= \sum_{i=1}^{2n} i^2 \\
   &= \frac{2n(2n+1)(4n+1)}{6} \in \Theta(n^3),
   \end{aligned}$$

   dove, per l'ultimo passaggio abbiamo usato la formula per il calcolo della somma dei primi $k$ quadrati, con $k = 2n$, e per la sommatoria $(*)$ abbiamo osservato che i termini della sommatoria sono indipendenti dall'indice della sommatoria.

(d) for $i = 1$ to $2n$
  for $j = 1$ to $i$
    for $k = 1$ to $j$
      $x := x + 1$

*Soluzione.* Analogamente all'esercizio precedente scriviamo:

$$\begin{aligned}\sum_{i=1}^{2n}\sum_{j=1}^{i}\sum_{k=1}^{j}1 &= \sum_{i=1}^{2n}\sum_{j=1}^{i}j\\ &= \sum_{i=1}^{2n}\frac{i(i+1)}{2}\\ &= \frac{1}{2}\left(\sum_{i=1}^{2n}i^2+\sum_{i=1}^{2n}i\right)\\ &= \frac{1}{2}\left(\frac{2n(2n+1)(4n+1)}{6}+\frac{2n(2n+1)}{2}\right)\in\Theta(n^3).\end{aligned}$$

(e) $i := n$
  while $i \geq 1$
    $x := x + 1$
    $i := i/2$

*Soluzione.* L'istruzione viene eseguita una volta ad ogni iterazione e il numero di iterazioni è $\Theta(\log n)$.

(f) $j := n$
  while $j \geq 1$
    for $i = 1$ to $j$
      $x := x + 1$
    $j := j/2$

*Soluzione.* Fissato $j$, istruzione viene eseguita $j$ volte ad ogni iterazione del ciclo esterno. I valori che $j$ può assumere dipendono dal ciclo esterno e sono $n, n/2, n/4, n/8, \ldots 2, 1$. Quindi il numero che cerchiamo è:

$$\sum_{i=0}^{\log n}\frac{n}{2^i} = n\sum_{i=0}^{\log n}\frac{1}{2^i} = n\cdot\frac{1-(\frac{1}{2})^{\log n+1}}{1-\frac{1}{2}} = n\cdot 2\cdot\underbrace{\left(1-\left(\frac{1}{2}\right)^{\log n+1}\right)}_{=\alpha},$$

dove abbiamo utilizzato la formula della somma parziale $k$-esima per le serie geometriche, per $k = \log n$. Abbiamo

$$\frac{1}{2}\leq\alpha = 1-\left(\frac{1}{2}\right)^{\log n+1}\leq 1 \text{ per } n\geq 1,$$

quindi $\alpha\in\Theta(1)$ e

$$\sum_{i=0}^{\log n}\frac{n}{2^i} = n\cdot 2\alpha\in\Theta(n).$$

**IN GENERALE** valgono i seguenti risultati:

- *Somma dei primi $n$ numeri naturali:*
$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \in \Theta(n^2).$$
- *Somma dei primi $n$ quadrati:*
$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \in \Theta(n^3).$$
- *Somma parziale $n$-esima della serie geometrica:* per $x \neq 1$ abbiamo che
$$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}.$$
  - Quando $0 < x < 1$ abbiamo che
  $$1 = x^0 \leq \sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x} \leq \frac{1}{1-x} \text{ e quindi } \sum_{i=0}^{n} x^i \in \Theta(1).$$
  La disuguaglianza di sinistra vale perchè la somma parziale è non più piccola del suo primo termine (quello con $i = 0$). La disuguaglianza di destra vale perchè $1 - x^{n+1} \leq 1$. Il valore $1/(1-x)$ è una costante perchè non dipende da $n$.
  - Quando $x = 2$ abbiamo
  $$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x} = \frac{1-2^{n+1}}{-1} = 2^{n+1} - 1 \in \Theta(2^n).$$
  - Quando $x > 1$ abbiamo
  $$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x} = \frac{1}{x-1}\left(x^{n+1} - 1\right) \in \Theta(x^n).$$

5. *Esercizio.* **[Esercizio da Prima Parte del compito]** Dire quale valori stampano i seguenti frammenti di codici.

   **N.B.:** Nell'istruzione `for` lo `step` indica l'incremento della variabile di controllo del ciclo ad ogni iterazione. Di default è uno, ma può essere modificato. Per esempio se abbiamo "`for i=1 to n step 2`" la variabile i assumerà i valori 1, 3, 5, 7, 9 e così via fino ad arrivare a n nel caso in cui n sia dispari o n-1 nel caso in cui n sia pari.

   (a) $c := 0$
   for $i = 2$ to $24$ step $2$
     for $j = 1$ to $24$ step $2$
       $c := c + 1$
   print $c$

   *Soluzione.* Il valore di $c$ che viene stampato è uguale al numero di volte in cui l'istruzione $c := c + 1$ viene eseguita, perchè il valore iniziale di $c$ è zero e viene incrementato di uno ogni volta che l'istruzione viene eseguita.
   Il ciclo più interno esegue l'istruzione $12$ volte (una volta per ogni numero dispari tra $1$ compreso e $24$) e questo ciclo viene eseguito $12$ volte da (una volta per ogni numero pari tra $2$ compreso e $24$ compreso) quello esterno. Quindi, abbiamo che
   $$c = 12 \cdot 12 = 144.$$

(b) $c := 0$
for $i = 2$ to $27$ step $1$
  for $j = i + 1$ to $28$
    $c := c + 1$
print $c$

*Soluzione.*

$$c = \sum_{i=2}^{27} \underbrace{\sum_{j=i+1}^{28} 1}_{(a)} = \sum_{i=2}^{27} \underbrace{(28 - (i+1) + 1)}_{=28-i} = \underbrace{\sum_{i=2}^{27} 28}_{(b)} - \underbrace{\sum_{i=2}^{27} i}_{(a)} = 28 \cdot 26 - \left[ \frac{27 \cdot 28}{2} - 1 \right] = 728 - 378 + 1 = 351.$$

Infatti, nella sommatoria $(a)$ sommiamo $1$ per ogni $j$ tra $i+1$ (incluso) e $28$ (incluso), ovvero $28-(i+1)+1 = 28 - 1$ volte. Nelle sommatorie $(b)$ e $(c)$ bisogna fare attenzione perché l'indice parte da parte da $2$: in $(b)$ il $28$ viene sommato $27 - 2 + 1 = 26$ volte, mentre $(c)$ non comprende l'$1$ (che va sottratto dalla somma dei primi $27$ numeri naturali).

(c) $c := 0$
for $i = 0$ to $25$ step $1$
  for $j = i + 1$ to $25$ step $1$
    $c := c + 1$
print $c$

*Soluzione.*

$$\sum_{i=0}^{25} \sum_{j=i+1}^{25} 1 = \sum_{i=0}^{25} (25 - i) = \underbrace{\sum_{i=0}^{25} 25}_{(a)} - \sum_{i=0}^{25} i = 25 \cdot 26 - \frac{25 \cdot 26}{2} = \frac{25 \cdot 26}{2} = 325,$$

facendo attenzone che l'indice della sommatoria esterna parte da zero, quindi in $(a)$ il $25$ viene sommato $26$ volte.

(d) $c := 0$
for $i = 0$ to $26$ step $2$
  for $j = i$ to $26$ step $2$
    $c := c + 1$
print $c$

*Diversi modi di arrivare alla soluzione:*

- Il ciclo esterno viene eseguito una volta per ogni numero pari tra $0$ e $26$ (estremi compresi). Il ciclo interno viene eseguito una volta per ognuno dei valori precedenti. Sia $j$ uno di tali valori: è un numero pari e il numero di numeri pari tra $j$ e $26$ (estremi compresi) è $\frac{26-j}{2} + 1 = 14 - \frac{j}{2}$. Poiché $j$ assume tutti i valori pari da $0$ a $26$, la sua metà $\frac{j}{2}$ assume tutti i valori (pari e dispari) tra $0$ e $13$. Quindi il numero di volte in cui il ciclo interno esegue l'istruzione è $14$ la prima volta, $13$ la seconda, $12$ la terza, $\ldots$ ,$1$ la quattordicesima volta. In conclusione:

$$c = \sum_{k=1}^{14} 1 = \frac{14 \cdot 15}{2} = 105.$$

- Il ciclo esterno viene eseguito solo per i numeri pari compresi tra 0 e 26, ovvero per tutti i numeri della forma $2k$, per $k \in \{0, 1, 2, 3, \ldots, 13\}$. Quello interno, fissato $i = 2k$, per un certo $k$, per tutti i numeri pari compresi tra $2k$ e 26, ovvero per tutti i numeri della forma $2t$, per $t \in \{k, k+1, \ldots, 13\}$. Per ogni esecuzione del ciclo interno $c$ viene incrementato di 1. Quindi, possiamo scrivere:

$$\sum_{k=0}^{13} \sum_{t=k}^{13} 1 =^{(1)} \sum_{k=0}^{13} (13 - k + 1) =^{(2)} \sum_{k=0}^{13} 14 - \sum_{k=0}^{13} k =^{(3)} 14 \cdot 14 - \frac{13 \cdot 14}{2} = 196 - 91 = 105.$$

  Infatti:
  (1) La sommatoria $\sum_{k=0}^{13} 1$ somma 1 per ogni intero compreso tra $k$ e 13, estremi compresi. Quindi il risultato è $13 - k + 1 = 14 - k$;
  (2) Abbiamo *spezzato* in due la sommatoria $\sum_{k=0}^{13}(14 - k) = \sum_{k=0}^{13} 14 - \sum_{k=0}^{13} k$;
  (3) La sommatoria $\sum_{k=0}^{13} 14$ somma 14 er ogni intero compreso tra 0 e 13, estremi compresi. Quindi il risultato è $14 \cdot (13 - 0 + 1) = 14 \cdot 14$. Inoltre, la sommatoria $sum_{k=0}^{13} k$ non è altro che la somma dei primi 13 numeri naturali, quindi il risultato è $(13 \cdot 14)/2$.
- In alternativa (più empirica che analitica) provare stabilire tutti i valori assunti da $i$ e $j$ simulando l'esecuzione del codice.

(e) $c := 0$
  for $i := 0$ to $20$ step $1$ do
    $j := 0$
    while $i + j \leq 21$ do
      $j := j + 1$
      $c := c + 1$
  print $c$

*Soluzione.* Iniziamo ad analizzare cosa succede nel ciclo più interno al variare del valore di $i$:

- quando $i = 0$, il ciclo `while` viene eseguito una volta per ogni intero da $0$ (alla prima iterazione abbiamo anche $j = 0$) a $21$ (e alla fine abbiamo $j = 21$). Quindi la variabile $c$ viene incrementata 22 volte (una per ogni valore assunto da $j$. **Attenzione**, il primo valore di $j$ è zero e non 1, per questo i valori diversi assunti da $j$ sono 22 e non 21).
- quando $i = 1$, il ciclo `while` viene eseguito una volta per ogni intero da $1$ (alla prima iterazione abbiamo sempre che $j = 0$) a $21$ (e alla fine abbiamo $j = 21 - 1 = 20$). Quindi la variabile $c$ viene incrementata $21$ volte.
- quando $i = 2$, il ciclo `while` viene eseguito una volta per ogni intero da $2$ (alla prima iterazione abbiamo sempre che $j = 0$) a $21$ (e alla fine abbiamo $j = 21 - 2 = 19$). Quindi la variabile $c$ viene incrementata $20$ volte.

E così via. Ad una generica iterazione, quando $i = k$ per qualche $k$ compreso tra $0$ e $20$ avremo che

- quando $i = k$, il ciclo `while` viene eseguito una volta per ogni intero da $k$ (alla prima iterazione abbiamo sempre che $j = 0$) a $21$ (e alla fine abbiamo $j = 21 - k = 21 - i$). Quindi la variabile $c$ viene incrementata $21 - i + 1$ volte.

L'ultima volta che il ciclo `while` viene eseguito è per $i = 20$ e la variabile $c$ viene incrementata $21 - i + 1 = 21 - 20 + 1 = 2$ volte.

Poiché l'incremento di $c$ è di $1$, allora il numero totale degli incrementi ci dà anche il valore finale di $c$ (che è inizializzata a zero). In conclusione abbiamo

$$c = \sum_{t=2}^{22} t = \sum_{t=1}^{22} t - 1 = \frac{22 \cdot 23}{2} = 253 - 1 = 252.$$

(f) $c := 0$
for $i := 0$ to $26$ step $1$ do
    $j := 0$
    while $i + j \leq 27$ do
        $j := j + 1$
        $c := c + 1$
print $c$

*Soluzione.* Ragionando in modo analogo all'esercizio precedente abbiamo che $c = 405$.

(g) $c := 0$
for $i := 0$ to $27$ step $2$ do
    $j := 0$
    while $i + j \leq 28$ do
        $j := j + 1$
        $c := c + 1$
print $c$

*Soluzione.* Possiamo ragionare in modo analogo all'esercizio precedente, ma questa volta dobbiamo fare attenzione perché lo step è 2 e non uno. Quindi

- quando $i = 0$, il ciclo `while` viene eseguito una volta per ogni intero da $0$ (alla prima iterazione abbiamo anche $j = 0$) a $28$ (e alla fine abbiamo $j = 28$). Quindi la variabile $c$ viene incrementata $29$ volte.
- quando $i = 2$, il ciclo `while` viene eseguito una volta per ogni intero da $2$ (alla prima iterazione abbiamo sempre che $j = 0$) a $26$ (e alla fine abbiamo $j = 28 - 2 = 26$). Quindi la variabile $c$ viene incrementata $27$ volte.
- quando $i = 4$, il ciclo `while` viene eseguito una volta per ogni intero da $4$ (alla prima iterazione abbiamo sempre che $j = 0$) a $24$ (e alla fine abbiamo $j = 28 - 4 = 24$). Quindi la variabile $c$ viene incrementata $25$ volte.

E così via. Ad una generica iterazione, $i$ sarà un numero pari compreso tra $0$ e $26$. Infatti il valore successivo di $i$ sarebbe $28$ che è fuori dal range e non viene eseguita una nuova iterazione del `for` per questo valore di $i$. Quindi, possiamo scrivere $i = 2k$ per qualche $k$ compreso tra $0$ e $13$ avremo che

- quando $i = 2k$, il ciclo `while` viene eseguito una volta per ogni intero da $2k$ (alla prima iterazione abbiamo sempre che $j = 0$) a $28$ (e alla fine abbiamo $j = 28 - 2k$). Quindi la variabile $c$ viene incrementata $28 - 2k + 1 = 29 - 2k$ volte.

L'ultima iterazione sarà per $k = 13$ con un incremento di $c$ pari a $29 - 2 \cdot 13 = 3$.
Abbiamo almeno due modi per arrivare al risultato:

1. Ci accorgiamo che l'incremento di $c$ è uguale alla somma dei numeri dispari che vanno da $3$ (incluso) a $29$ (incluso). Scriviamo un numero dispari generico come $2j + 1$. A noi interessano solo quelli per cui $j$ va da $1$ a $14$ e abbiamo:

$$\sum_{j=1}^{14} (2j + 1) = 2 \sum_{j=1}^{14} j + \sum_{j=1}^{14} 1 = 2 \cdot \frac{14 \cdot 15}{2} + 14 = 210 + 14 = 224.$$

2. Continuiamo con il ragionamento di prima:

$$c = \sum_{k=0}^{13}(29 - 2k) = \sum_{k=0}^{13} 29 - 2\sum_{k=0}^{13} k = 29 \cdot 14 - 2 \cdot \frac{13 \cdot 14}{2} = 406 - 182 = 224.$$

(h) $c := 0$
for $i := 0$ to $28$ step $1$ do
    $j := 0$
    while $i + j \leq 28$ do
        $j := j + 1$
        $c := c + 1$
print $c$

*Soluzione.* Ragionando in modo analogo agli esercizi precedenti e simili abbiamo che $c = 435$. Bisogna fare attenzione che, in questo caso, l'ultima iterazione del ciclo `for` si ha per $i = 28$ e questo porta ad un'ultima esecuzione del ciclo interno `while` in cui ci sarà una sola iterazione, quindi

$$c = \sum_{t=1}^{29} t = \frac{29 \cdot 30}{2} = 435.$$