INFORMATICA GENERALE - I-Z

Claudia Malvenuto - Daniele A. Gewurz
Scheda esercizi n. 2

1. Esaminare i seguenti algoritmi e per ognuno dare una stima del numero di volte che viene stampata la stringa "Hello". (Può essere difficile calcolare il numero esatto, ma è sufficiente calcolarne un Θ). Per farlo, è conveniente per ogni ciclo calcolare quante volte viene eseguito quel ciclo e che contributo dà ogni iterazione.

Algoritmo 1 Hello1 (intero n)

```
i ← 27
while i ≤ n do
    print "Hello"          ▷ Quante volte viene eseguita questa istruzione?
    i ← i * 3
end while
```

Algoritmo 2 Hello1 (intero n)

```
for i = 1 to n do
    for j = 1 to i do
        for k = 1 to j do
            print "Hello"
        end for
    end for
end for
```

Algoritmo 3 Hello3 (intero n)

```
while n ≥ 1 do
    j ← 0
    while j ≤ n do
        print "Hello"
        j ← j + 4
    end while
    n ← n/4
end while
```

2. Scrivere un algoritmo ricorsivo per il calcolo del fattoriale, usandone la definizione ricorsiva che dà la base $1! = 1$ e la relazione di ricorrenza $n! = (n-1)! \cdot n$. Calcolarne la complessità.

3. Scrivere una versione ricorsiva per la ricerca lineare in un vettore, e calcolare il corrispondente tempo di esecuzione (caso migliore e peggiore).

4. Sia A un array di n interi, inizializzati tutti a 0. Progettare un algoritmo che memorizzi in $A[i]$ l'i-esimo numero di Fibonacci F_i per ogni i in $\{1, \ldots, n\}$. L'algoritmo deve operare ricorsivamente secondo la seguente strategia:

a) se $i = 1$ o $i = 2$, pone $A[i] = 1$ ed esce;

b) se $A[i-1] = 0$, calcola $A[i-1]$ ricorsivamente;

c) se $A[i-2] = 0$, calcola $A[i-2]$ ricorsivamente;

d) memorizza in $A[i]$ la somma $A[i-1] + A[i-2]$ ed esce.

- Scrivere lo pseudocodice dell'algoritmo.
- Disegnare l'albero della ricorsione per il calcolo di F_6.
- Analizzare il tempo di esecuzione dell'algoritmo.

Il procedimento usato in un algoritmo di questo tipo è detto *memoizzazione*: durante l'esecuzione l'algoritmo prende degli "appunti" (*memo* in inglese) per non dover ricalcolare più volte uno stesso dato (in questo caso i numeri di Fibonacci già calcolati).

5. Si consideri l'algoritmo in pseudocodice per la ricerca lineare:

Algoritmo 4 ricercaLineare (lista L, elem x) $\rightarrow$ booleano

```
for all y in L do
    if y = x then
        return "trovato"
    end if
    return "non trovato"
end for
```

Modificarlo in modo da risolvere questo problema:

Istanza: un vettore ordinato L di n interi e un numero intero x;

Soluzione: un intero $k \in \{1, 2, \ldots, n\}$ tale che $L[k] = x$, se esiste un intero con questa proprietà, e -1 in caso contrario.