

1 Esercizio 1

Input

- Numero intero $N \geq 1$

Output

- Numero di coppie (i, j) tali che i e j compresi tra 1 e N

Codice di base

```
function Count_1(N: Int)
  sum = 0
  for (i = 1; i <= N; i = i + 1)
    for (j = i; j <= N; j = j + 1)
      sum = sum + 1
  return sum
```

Tempo di esecuzione:

$$2 + 3N + 1 + \sum_{i=1}^N [3(N + 1 - i)] + 2 \sum_{i=1}^N (N + 1 - i)$$

Posso fare di meglio.

```
function Count_1(N: Int)
  sum = 0
  for (i = 1; i <= N; i = i + 1)
    sum = sum + (N + 1 - i)
  return sum
```

Tempo di esecuzione: $7N + 3$

Posso fare (ancora) meglio.

$$\sum_{i=1}^N (N + 1 - i) = \sum_{i=1}^N i = \frac{N(N + 1)}{2}$$

Quindi posso scrivere:

```
function Count_1(N: Int)
  return N * (N + 1) / 2
```

Tempo di esecuzione: 5

2 Esercizio 2

```
var A: [Int] = un array contenente interi;
var res: Bool = false;
const n: Int = A.length;
for(var k: Int = 0; k < sqrt(n); k++)
  for (var h: Int = 0; h < k; h++)
    for (var t: Int = 0; t < h; t++)
      res = res || ((A[t] % 2 == 0) && (A[t+1] % 2 == 1))
```

Dichiarazioni

Analisi statica

$$\Delta = \langle (A, [Int]Loc), (res, BoolLoc), (n, Int) \rangle$$

Analisi dinamica

Stato 0

$$D_1; D_2; D_3; C$$

$$\rho = \emptyset$$

$$\sigma = \emptyset$$

Stato 1

$$D_2; D_3; C$$

$$\rho = \langle (A, [Loc]) \rangle$$

$$\sigma = \langle ([Loc], [a_1, a_2, \dots]) \rangle$$

Stato 2

$$D_3; C$$

$$\rho = \langle (A, [Loc]), (res, L_0) \rangle$$

$$\sigma = \langle ([Loc], [a_1, a_2, \dots]), (L_0, false) \rangle$$

Complessità

Il for più esterno viene eseguito \sqrt{n} volte

Il for intermedio è nell'ordine di $O(n)$ perchè tramite la formula di Gauss:

$$O(\sqrt{n}^2) = \frac{\sqrt{n}(\sqrt{n} + 1)}{2}$$

Il for più interno è dell'ordine $O(n\sqrt{n})$

$$h = \frac{k(k+1)}{2}$$

$$k = \frac{\sqrt{n}(\sqrt{n} + 1)}{2}$$

$$\begin{aligned}
s &= \frac{h(h+1)}{2} \\
s &= \frac{\frac{k(k+1)}{2} \left(\frac{k(k+1)}{2} + 1 \right)}{2} \\
s &= \frac{\frac{\frac{\sqrt{n}(\sqrt{n}+1)}{2} \left(\frac{\sqrt{n}(\sqrt{n}+1)}{2} + 1 \right)}{2} \left(\frac{\frac{\sqrt{n}(\sqrt{n}+1)}{2} \left(\frac{\sqrt{n}(\sqrt{n}+1)}{2} + 1 \right)}{2} + 1 \right)}{2}
\end{aligned}$$

3 Esercizio 3

Verificare che il programma sia corretto dal punto di vista di semantica statica.

```

var xG:Double = 3.5;
if (xG < 3.5) {
    const xL:Double = 13.2;
    var sommaIf:Double = xG + xL;
}
else {
    var sommaElse:Double = xG + xL;
}
sommaIf = sommaIf + 1;
sommaElse = sommaElse + 1;

```

1. $D_1 =$ var xG:Double = 3.5;
2. $D_2 =$ const xL:Double = 13.2;
3. $D_3 =$ var sommaIf:Double = xG + xL;
4. $D_4 =$ var sommaElse:Double = xG + xL;
5. $C_1 =$

```

const xL:Double = 13.2
var sommaIf:Double = xG + xL;

```

6. $C_2 =$ var sommaElse:Double = xG + xL;
7. $C_3 =$

```

if (xG < 3.5) {
    const xL:Double = 13.2;
    var sommaIf:Double = xG + xL;
}

```

```

else {
  var sommaElse:Double = xG + xL;
}

```

8. $C_4 = \text{sommaIf} = \text{sommaIf} + 1;$

9. $C_5 = \text{sommaElse} = \text{sommaElse} + 1;$

10. $C_6 = C_3; C_4; C_5$

11. $C_7 = C_4; C_5$

$$\frac{\frac{\emptyset \vdash_e 3.5: Double}{\emptyset \vdash_d D_1: \Delta_1}, \quad \frac{\Delta_1 \vdash_c C_3, \quad \frac{\Delta_1 \vdash_c C_4, \Delta_1 \vdash_c C_5}{\Delta_1 \vdash_c C_7}}{\Delta_1 \vdash_c C_6}}{\emptyset \vdash_c P}$$