

## TEMPO DI ESECUZIONE

Il tempo di esecuzione di un programma può dipendere da numerosi fattori (compilatore usato, mole d'input etc...).

Questo può essere studiato più estrattamente tramite un modello computazionale.

## MODELLO COMPUTAZIONALE RAM

È una macchina che può eseguire qualsiasi tipo di algoritmo, questa in particolare a memoria principale infinita (celle infinite che contengono un numero finito di dati) che impiega lo stesso tempo ad accedere ad ogni cella.

Funziona grazie a un solo processore + programma.

11 12 13 14  
22 23 24  
33 34  
44

## ANALISI ASINTOTICA

INPUT:  $N \geq 1$

OUTPUT: numero di coppie  $(i, j) \mid 1 \leq i \leq j \leq N$

Salviamo in una variabile  $c$  le coppie valide

```
int Count_0( int N)
1  sum = 0
2  for i =1 to N
3      for j =1 to N
4          if i <= j then
5              sum = sum+1
6  return sum
```

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

I costi delle righe sono

① 1

②  $2(N+1)$

③  $2N(N+1)$

④  $3N^2$

⑤  $\frac{N(N+1)}{2}$

⑥ 1

Sommando questi valori avremo la complessità

$$T_A(N) = 1 + 2N + 2 + \frac{5}{2}N(N+1) + 3N^2$$

$$= 4 + 2N + \frac{5}{2}N^2 + \frac{5}{2}N + 3N^2 = \frac{11}{2}N^2 + \frac{9}{2}N + 4$$

Abbiamo una crescita quadratica del tempo

FORMULE CHIUSE

Possiamo così ottimizzare:

```
int Count_1(int N)
```

```
1    sum = 0
```

```
2    for i = 1 to N
```

```
3        for j = i to N
```

```
4            sum = sum + 1
```

```
5    return sum
```

Il cui costo è  $\frac{3}{2}N^2 + \frac{13}{2}N + 4$

Notiamo come la riga 4 dipende da  $i$  quindi fissato  $i$  già sappiamo che sarà eseguita  $N+1-i$  volte

Possiamo quindi ridurci a questa forma:

```

int Count_2(int N)
1   sum = 0   _____ 1
2   for i = 1 to N   _____ 2N+2
3       sum = sum + (N+1-i)   _____ 4N
4   return sum   _____ 1

```

Il cui costo è  $6N+4$

Ma ciò che fa l'algoritmo è eseguire  $N$  volte la riga 3 quindi:

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

Quindi l'algoritmo si riduce a

```

int Count_3(int N)
1   sum = N * (N+1) / 2   _____ 4
2   return sum   _____ 1

```

Il cui costo è semplicemente 5.

