

Esempio

Specifica:

- un programma che, dato un numero n , dice se è **primo** oppure no

*un numero è primo
se è divisibile solo per 1 e per se stesso
cioè
se non è multiplo di nessun numero*

es. 11 è primo, 7 è primo
12 non è primo infatti $12:3=4$ con resto 0
130 non è primo infatti $130:10=13$ con resto 0

S. Crafa A.A. 23/24

Esempio

Specifica:

- un programma che, dato un numero n , dice se è **primo** oppure no

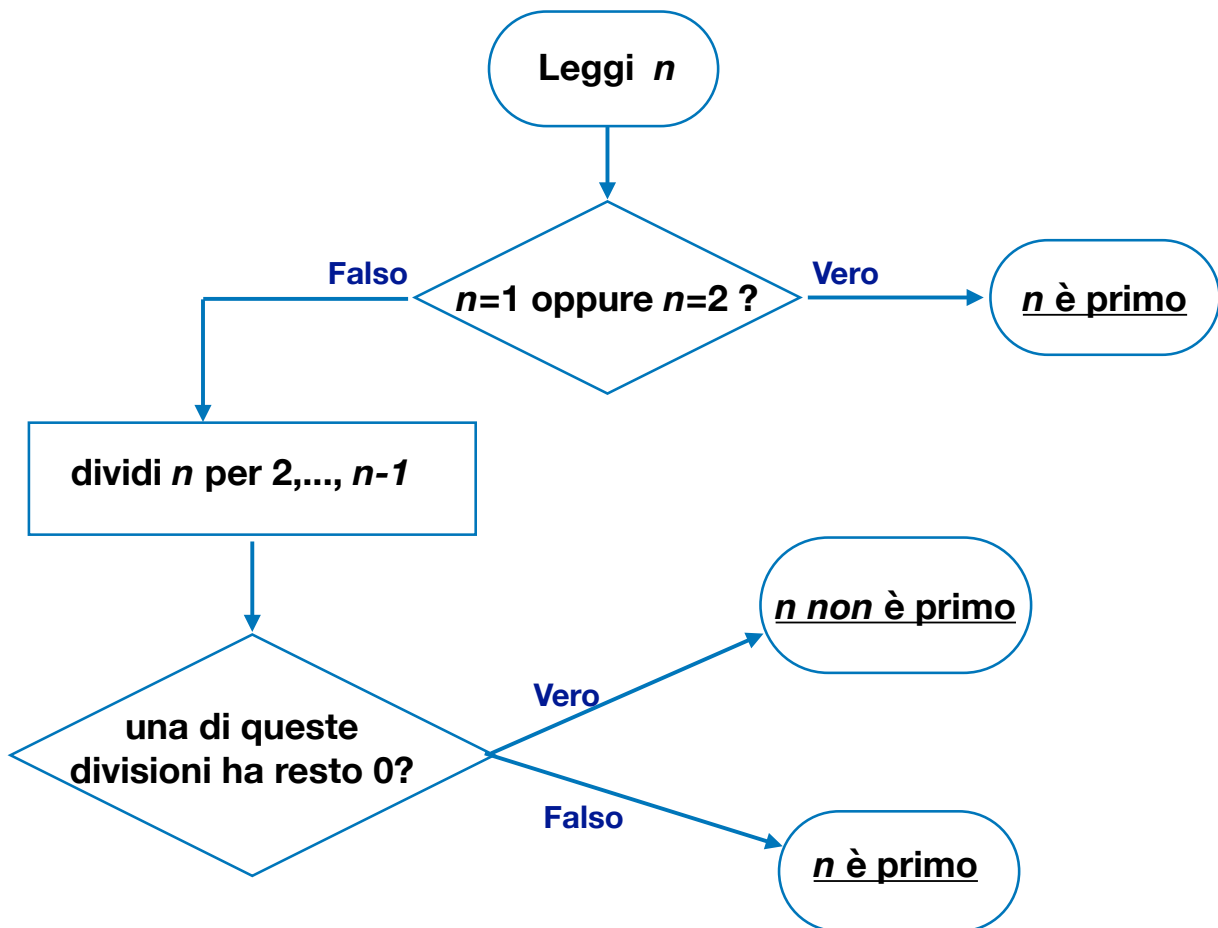
Algoritmo:

INPUT : un numero n

OUTPUT : " n è primo" opp " n non è primo"

- se** n è il numero 1 oppure 2, **allora** n è primo
- altrimenti**:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n-1$
 - se** una di queste divisioni ha resto 0 allora dici " n non è primo"
 - altrimenti** (cioè tutte le divisioni hanno resto non 0) dici " n è primo"

S. Crafa A.A. 23/24



S. Crafa A.A. 23/24

Esempio

Specifica:

- un programma che, dato un numero n , dice se è primo oppure no

Algoritmo:

INPUT : un numero n **OUTPUT** : " n è primo" opp " n non primo"

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n-1$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto diverso da 0) allora n è primo

è ok dividere fino a $n-1$? $n/2$?
è ok separare i casi $n=1,2$?

dato l'algoritmo posso **ragionare** sulla **correttezza**
della procedura risolutiva

quando mi sembra di avere un algoritmo corretto
devo **scrivere un software che lo esegue**

Esempio: Algoritmo che, dato un numero n dice se è primo oppure no

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n-1$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto non 0) n è primo

che istruzioni software gli faccio corrispondere?

- faccio *tutte* le divisioni e poi guardo *tutti* i risultati?
- ma se $n=100$, divido per 2,3,4,...,99 ma potrei accorgermi subito che già la divisione per 2 ha resto 0, evitando altre 98 divisioni inutili !
- scelgo l'istruzione:
"dividi n per div e controlla il resto, se il resto è 0 smetti, altrimenti ripeti dividendo per $div+1$. Inizia con $div=2$ e smetti di ripetere se $div = n-1$ "

S. Crafa A.A. 23/24

Esempio: Algoritmo che, dato un numero n dice se è primo oppure no

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n-1$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto non 0) n è primo

```
n = int(input())
if n==1 or n==2:
    print("n primo")
else:
    div = 2
    resto = n%div
    while (resto != 0 and div < n-1):
        div = div+1
        resto = n%div
    if resto == 0:
        print("n non primo")
    else:
        print("n primo")
```

"dividi n per div e controlla il resto, se il resto è 0 smetti, altrimenti ripeti dividendo per $div+1$. Smetti di ripetere se $div=n-1$ "

S. Crafa A.A. 23/24

Esempio: Algoritmo che, dato un numero n dice se è primo oppure no

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n-1$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto non 0) n è primo

l'algoritmo è corretto
ma
l'implementazione ha
un errore di semantica

es. 8--> "n non primo"
7--> "n non primo"

```
n = int(input())
if n==1 or n==2:
    print("n primo")
else:
    div = 1
    resto = n%div
    while (resto != 0 and div < n-1):
        div = div+1
        resto = n%div
    if resto == 0:
        print("n non primo")
    else:
        print("n primo")
```

errore!

S. Crafa A.A. 23/24

Esempio: Algoritmo che, dato un numero n dice se è primo oppure no

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, fino a dividere n per $n/2$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto non 0) n è primo



- L'implementazione richiede di definire (**disambiguare**) tanti dettagli dell'algoritmo
- Più difficile se chi ha scritto l'algoritmo è diverso da chi scrive il software!

```
n = int(input())
if n==1 or n==2:
    print("n primo")
else:
    div = 1
    resto = 1
    while (resto != 0 and div <= n/2):
        div=div+1
        resto = n%div
    if resto == 0:
        print("n non primo")
    else:
        print("n primo")
```

corretto ?!

S. Crafa A.A. 23/24

diversi livelli di astrazione

un **errore** può stare
su ciascuno dei livelli !!



1. **Specifica** del programma: *"Il programma prende in input un numero e ritorna in output se è primo oppure no"*



2. **Algoritmo** risolutivo:

- se n è 1 oppure 2, allora n è primo
- altrimenti:
 - dividi n per 2, poi per 3, poi per 4, ..., fino a dividere n per $n/2$
 - se una di queste divisioni ha resto 0 allora n non è primo
 - altrimenti (cioè tutte le divisioni hanno resto non 0) n è primo



3. **Implementazione**

```
n = int(input())
if n==1 or n==2:
    print("n primo")
else:
    div = 2
    resto = n%div
    while (resto != 0 and div < n-1):
        div=div+1
        resto = n%div
    if resto == 0:
        print("n non primo")
    else: print("n primo")
```

Sistema Operativo

Hardware

S. Crafa A.A. 23/24

app di tracciamento contatti in pandemia



1. **Specifica** del programma: *"Il programma prende in input i contatti ... calcola ... manda il risultato"*



2. **Algoritmo** risolutivo:

- attiva il bluetooth low energy,
- genera stringa casuale ogni 5 sec e invia
- controlla se arriva una stringa e memorizza
- ...



3. **Implementazione**



Sistema Operativo

Hardware



S. Crafa A.A. 23/24

diversi livelli di astrazione



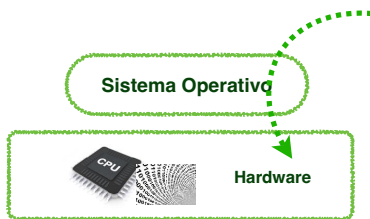
1. **Specifica** del programma:



2. **Algoritmo** risolutivo:



3. **Implementazione**



Twitter ---> **X**?

UniWeb?

cercare esempi di
applicazioni in cui
queste distinzioni vi
sembrano interessanti

=> *Forum*

S. Crafa A.A. 23/24

App Immuni

<https://www.immuni.italia.it/>

- **High level description**

<https://github.com/immuni-app/immuni-documentation>

[https://github.com/immuni-app/immuni-dashboard-data/blob/master/dati/
andamento-settimanale-dati-regionali-latest.csv](https://github.com/immuni-app/immuni-dashboard-data/blob/master/dati/andamento-settimanale-dati-regionali-latest.csv)

- **documentazione tecnica**

<https://github.com/immuni-app/immuni-app-android>

<https://github.com/immuni-app/immuni-documentation/blob/master/Technology.md>

- **esempi di codice sorgente** (Kotlin e Go)

[https://github.com/immuni-app/immuni-app-android/blob/development/app/src/main/java/it/
ministerodellasalute/immuni/logic/exposure/repositories/ExposureAnalyticsStoreRepository.kt](https://github.com/immuni-app/immuni-app-android/blob/development/app/src/main/java/it/ministerodellasalute/immuni/logic/exposure/repositories/ExposureAnalyticsStoreRepository.kt)

[https://github.com/immuni-app/immuni-app-android/blob/development/app/src/main/java/it/
ministerodellasalute/immuni/logic/notifications/AppNotificationManager.kt](https://github.com/immuni-app/immuni-app-android/blob/development/app/src/main/java/it/ministerodellasalute/immuni/logic/notifications/AppNotificationManager.kt)

<https://github.com/google/exposure-notifications-server/blob/main/internal/integration/client.go>

diversi livelli di astrazione

1. **Specifica** del programma
2. **Algoritmo** risolutivo
3. **Implementazione**
4. **Infrastruttura di esecuzione**

- molto spesso i 3 livelli sono **confusi** e **mescolati**
- es. inizio *prototipo* di implementazione con idea grezza di cosa fa
 - se l'**algoritmo** resta **implicito**, rischio ci siano errori di correttezza
 - se la specifica (chiara) manca (es. solo commenti al codice), è difficile comunicare cosa fa il software, e mantenerlo



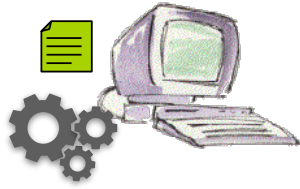
S. Crafa A.A. 23/24

diversi livelli di astrazione

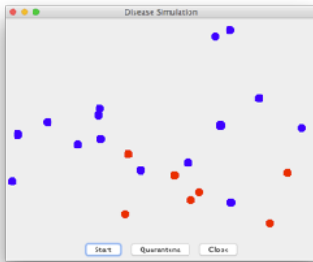
1. **Specifica** del programma
2. **Algoritmo** risolutivo
3. **Implementazione**
4. **Infrastruttura di esecuzione**

*spesso l'infrastruttura di esecuzione
ha importanti conseguenze
sui livelli superiori !*

dove avviene la computazione?



applicazione **locale**



```
print('Ciao! a cosa vuoi giocare?')
input('1.Scacchi 2.Indovina il numero 3.Guerra termonucleare globale')
print('Indovina il numero..Ottima scelta!')
answer = input('inserisci il numero: ')
if answer == 7 :
    print('Indovinato!')
else:
    print('Sbagliato...sarai piu fortunato in amore')
```

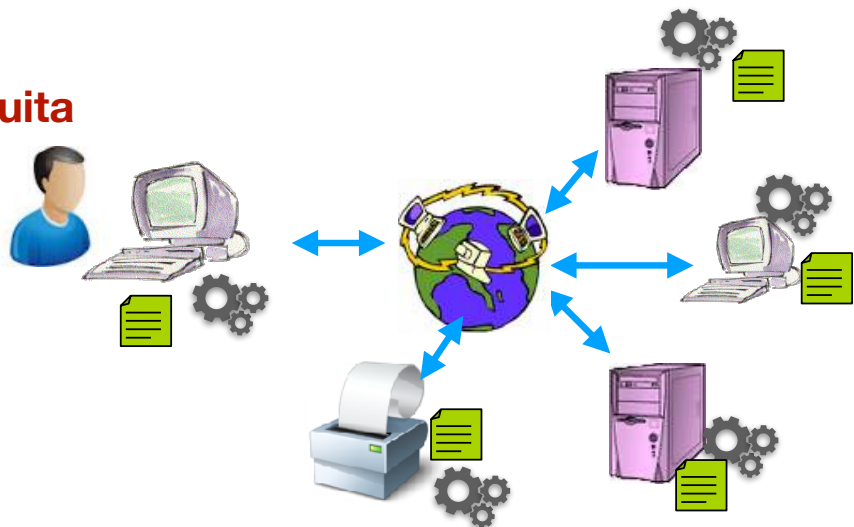
- dove si esegue il programma?
- se salvo/apro un progetto, dove risiede?

S. Crafa A.A. 23/24

dove avviene la computazione?

applicazione **distribuita**

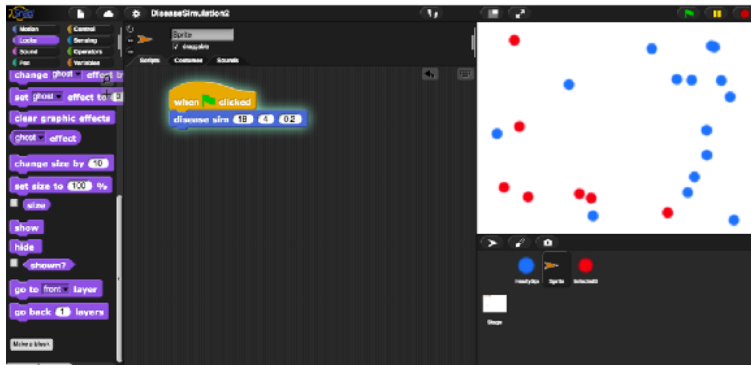
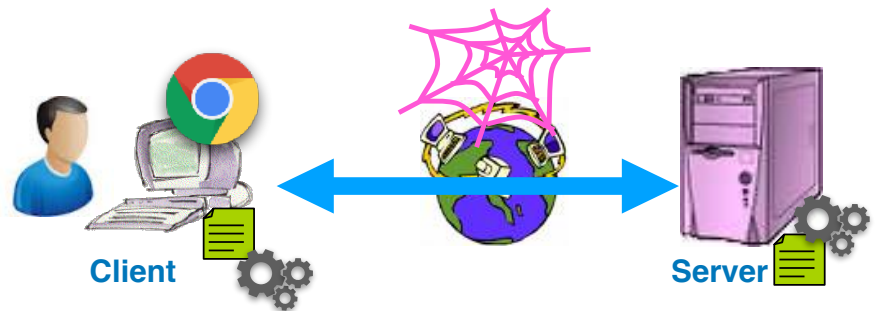
es. mail, stampare



S. Crafa A.A. 23/24

dove avviene la computazione?

applicazione Web
es. web mail, banking online,
twitter

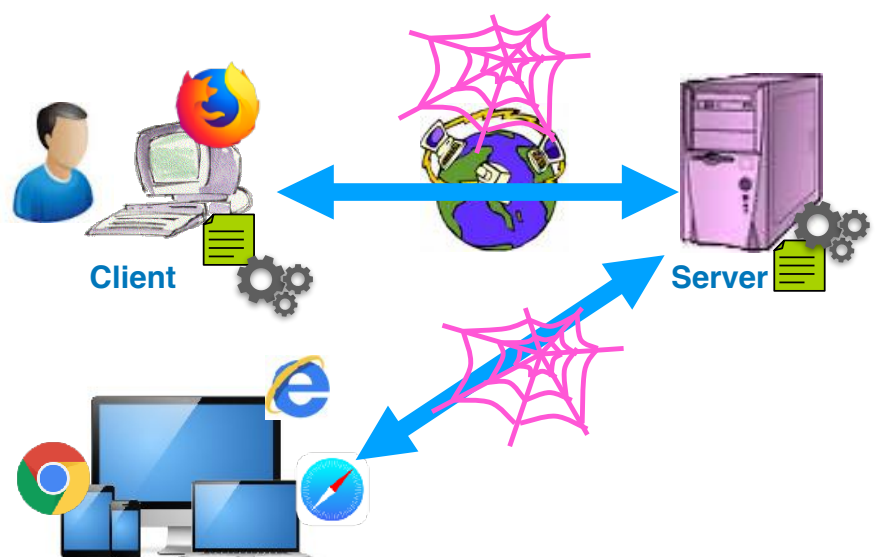


- dove esegue il programma?
- se salvo/apro un progetto, dove risiede?

S. Crafa A.A. 23/24

dove avviene la computazione?

applicazione Web
es. web mail, banking online,
Il Corriere



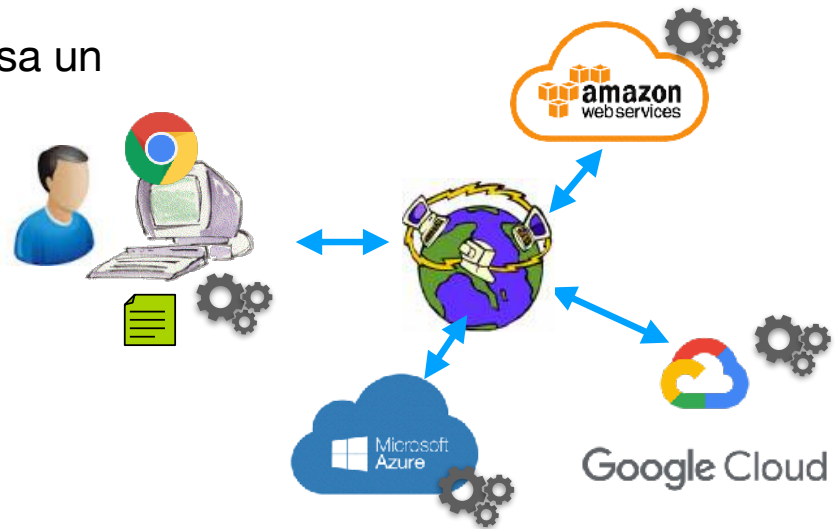
se un'applicazione funziona
tramite il browser web, la si
può usare da desktop, tablet,
smartphone

S. Crafa A.A. 23/24

dove avviene la computazione?

applicazione che usa un **servizio cloud**

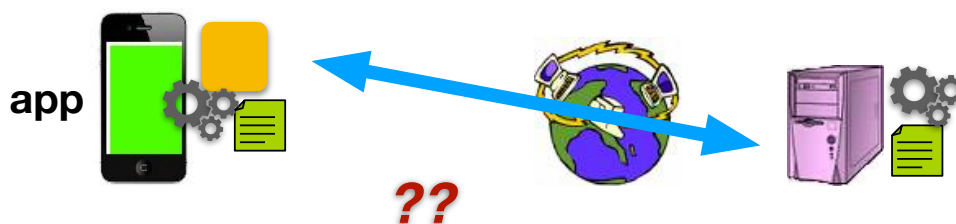
es. Moodle,
firma digitale,
archivio foto



cloud = un sistema distribuito privato e opaco

S. Crafa A.A. 23/24

dove avviene la computazione?



La app puo' causare computazione sia locale che remota

- se si connette a server remoto, **che dati gli manda?**
- **non è sempre facile capire** se c'è computazione remota (opp. disattivare rete)

c'è tanta complessità nascosta (*astrazione*) tra le istruzioni di un programma e l'esecuzione della macchina!
Bisogna esserne Consapevoli

S. Crafa A.A. 23/24

diversi livelli di astrazione

1. **Specifica** del programma
2. **Algoritmo** risolutivo
3. **Implementazione**
4. **Infrastruttura di esecuzione**



*spesso l'infrastruttura di esecuzione
ha importanti conseguenze
sui livelli superiori !*

- un **computer** (esecuzione *locale*)
- uno **smartphone** (esecuzione *parzialmente remota*)
- un **sistema distribuito** (esecuzione *su server tramite interfaccia web*)
- un **cloud** = un sistema distribuito privato e opaco

S. Crafa A.A. 23/24

diversi livelli di astrazione

1. **Specifica** del programma
2. **Algoritmo** risolutivo
3. **Implementazione**
4. **Infrastruttura di esecuzione**

- **Un errore può stare su ciascun livello:**
 - tenere presente che esistono tutti
 - *saper assegnare ad ogni errore il livello a cui appartiene*
- **Persone diverse si occupano di ciascun livello:**
 - ogni livello richiede competenze specifiche, dunque *responsabilità specifiche*
 - molto difficile comunicare informazioni (corrette) tra livelli, per molti motivi

S. Crafa A.A. 23/24

Algoritmo vs software

Esempio:

un programma che calcola la graduatoria degli insegnanti di scuola

1. L'algoritmo sottostante potrebbe utilizzare delle **regole di posizionamento discriminatorie**, fedelmente tradotte nel software.
 - In questo caso il **software** è corretto, mentre il problema sta nell'**algoritmo**.
 - **di chi è la responsabilità?**
 - la regola discriminatoria stava nella **specifica** del programma?
 - chi ha commissionato il software, **che informazioni/vincoli ha comunicato? In che formato** li ha comunicati?
 - se come **documentazione** abbiamo solo il **codice** del software, è difficile attribuire la responsabilità

S. Crafa A.A. 23/24

Algoritmo vs software

Esempio:

un programma che calcola la graduatoria degli insegnanti di scuola

2. Consideriamo inoltre il caso in cui l'**algoritmo/specifica non dia alcuna indicazione su come trattare i casi di pari merito**
 - cioè la **specifica** del programma (i.e. cosa deve fare) **non è completa**.
 - L'**algoritmo/specifica** quindi è **ambiguo**, ma il **software** che lo implementa è per definizione **non ambiguo**:
 - le istruzioni dei linguaggi di programmazione sono fatte in modo che **la macchina sappia come comportarsi in ogni situazione**,
 - **includere le situazioni a cui il programmatore non ha esplicitamente pensato!**

S. Crafa A.A. 23/24

sono (semanticamente) equivalenti o no a seconda di come è inizializzata la variabile **ammesso**, cioè **a seconda di come è scritto il resto del programma**

Porzione di codice che decide chi è ammesso oppure no ad una speciale posizione in graduatoria:

- `if (x > y) then ammesso = true`
- `if (x > y) then ammesso = true else ammesso = false`
- `if (x >= y) then ammesso = true else ammesso = false`

istruzioni molto simili ma che
possono produrre **conseguenze molto diverse!**

- Piccole differenze nelle istruzioni, come queste, possono **finire per gestire implicitamente i casi di pari merito**,
- anche **senza l'esplicita intenzione del programmatore**, che si è limitato ad implementare l'algoritmo senza accorgersi che la specifica non era completa.

S. Crafa A.A. 23/24

Pubblicato il 08/04/2019

N. 02270/2019REG.PROV.COLL.
N. 04477/2017 REG.RIC.



R E P U B B L I C A I T A L I A N A

IN NOME DEL POPOLO ITALIANO

Il Consiglio di Stato

in sede giurisdizionale (Sezione Sesta)

ha pronunciato la presente

SENTENZA

sul ricorso numero di registro generale 4477 del 2017, proposto da [...] contro

Ministero dell'Istruzione dell'Università e della Ricerca, Ufficio Scolastico Regione Puglia [...] per la riforma

della sentenza del T.A.R. per il Lazio, Sede di Roma, n. 12026 del 2016

Secondo gli appellanti, tale algoritmo avrebbe disposto i trasferimenti in una provincia piuttosto che in un'altra, in un posto di sostegno piuttosto che in un posto comune, senza tener conto delle preferenze indicate nelle rispettive domande di trasferimento, senza alcuna motivazione e in difetto della benché minima trasparenza

[...]

l'utilizzo di una procedura informatica che conduca direttamente alla decisione finale non deve essere stigmatizzata, ma anzi, in linea di massima, incoraggiata: essa comporta infatti numerosi vantaggi quali, ad esempio, la notevole riduzione della tempistica procedimentale per operazioni meramente ripetitive e **prive di discrezionalità**, l'esclusione di interferenze dovute a **negligenza (o peggior dolo)** del funzionario (essere umano) e la conseguente maggior garanzia di imparzialità della decisione automatizzata.

9 – Alla luce delle riflessioni che precedono, l'appello deve trovare accoglimento, sussistendo nel caso di specie la violazione dei principi di imparzialità, pubblicità e trasparenza, poiché non è dato comprendere per quale ragione le legittime aspettative di soggetti collocati in una determinata posizione in graduatoria siano andate deluse.

Infatti, l'impossibilità di comprendere le modalità con le quali, attraverso il citato algoritmo, siano stati assegnati i posti disponibili, costituisce di per sé un vizio tale da inficiare la procedura.

Non solo, gli esiti della stessa paiono effettivamente connotati dall'illogicità ed irrazionalità denunciate dalle appellanti, essendosi verificate **situazioni paradossali** per cui docenti con svariati anni di servizio si sono visti assegnare degli ambiti territoriali mai richiesti e situati a centinaia di chilometri di distanza dalla propria città di residenza