

Ingegneria del Software

A.A. 2005-06

**Università di Roma “La Sapienza”
Dipartimento di Informatica e Sistemistica**

**Soluzioni esercizi del compito
d’esame del 16 dicembre 2005**

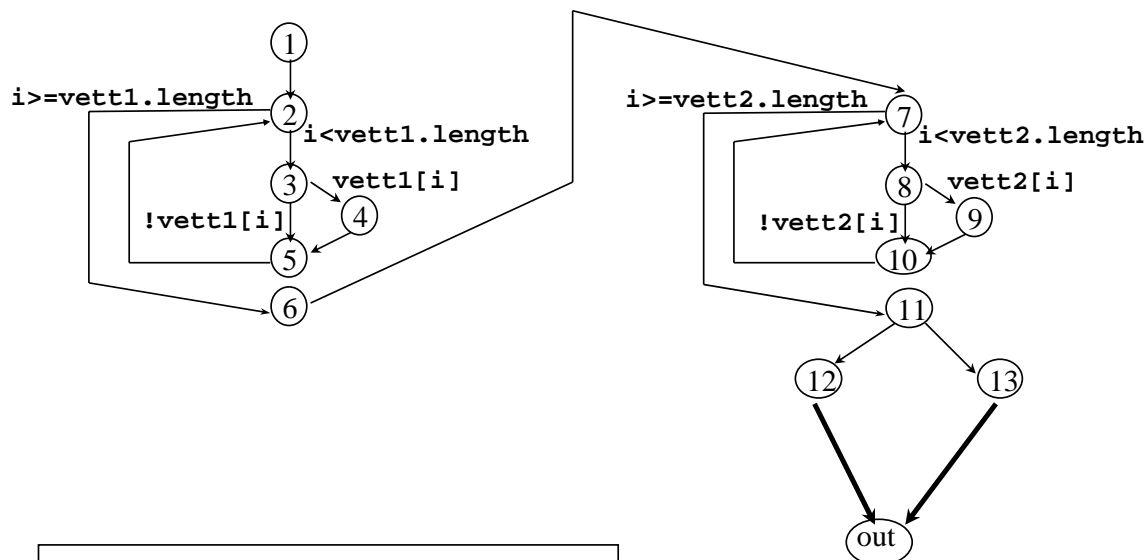
Soluzione esercizio 1

Passo 1: identificazione delle istruzioni

```
public static boolean conta(boolean[] vett1, boolean[] vett2) {  
  /* 1 */      int veri1 = 0, veri2= 0; int i = 0;  
  /* 2 */      while (i < vett1.length) {  
  /* 3 */          if (vett1[i])  
  /* 4 */              veri1++;  
  /* 5 */              i++;  
  }  
  /* 6 */      i = 0;  
  /* 7 */      while (i < vett2.length) {  
  /* 8 */          if (vett2[i])  
  /* 9 */              veri2++;  
  /* 10 */              i++;  
  }  
  /* 11 */      if (veri1 >= veri2)  
  /* 12 */          return true;  
  else  
  /* 13 */      return false;  
  } // conta
```

- Nel programma non esistono decisioni composte
- I cicli “for” vanno riscritti come cicli “while”, secondo la loro semantica

Passo 2: costruzione del grafo di controllo



Il grafo ha:

- 14 nodi
- 18 archi, che devono essere tutti coperti
- complessità ciclomatica pari a 6

Passo 3: progettazione casi di test

Caso di test	Variabile di input vett1[]	Variabile di input vett2[]	Output previsto conta
C1	{ }	{ }	true
C2	{ }	{ true }	false
C3	{ }	{ false }	true
C4	{ true }	{ }	true
C5	{ false }	{ }	true

Passo 4: individuazione dei cammini relativi ai casi di test

Cammini relativi a ciascun caso di test (come sequenza di nodi nel grafo di controllo):

- C1: 1-2-6-7-11-12-out
- C2: 1-2-6-7-8-9-10-7-11-13-out
- C3: 1-2-6-7-8-10-7-11-12-out
- C4: 1-2-3-4-5-2-6-7-11-12-out
- C5: 1-2-3-5-2-6-7-11-12-out

Passo 5: verifica della copertura degli archi

Arco	1-2	2-3	3-4	4-5	3-5	5-2	2-6	6-7	7-8	8-9	9-10	8-10	10-7	7-11	11-12	11-13	12-out	13-out
C1	x						x	x						x	x		x	
C2	x						x	x	x	x	x		x	x		x		x
C3	x						x	x	x			x	x	x	x		x	
C4	x	x	x	x		x	x	x						x	x		x	
C5	x	x			x	x	x	x						x	x		x	
Coperto	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Tutti gli archi sono coperti da almeno un caso di test
→ **abbiamo soddisfatto il criterio di copertura dei cammini**

Passo 6: verifica base

- Abbiamo proposto 5 casi di test
- La complessità ciclomatica è pari a 6
- Per definizione, una base è un insieme di cammini di cardinalità pari alla complessità ciclomatica
- Di conseguenza, l'insieme di casi di test proposti non può essere una base

Soluzione esercizio 3

Scelta delle azioni

- Il numero di errori per linee di codice sorgente indica che probabilmente l'azienda si trova ad un livello CMM non superiore a 1 (infatti a livello 2 si ha tipicamente un numero di errori per linea di codice non molto superiore a 1 su 1000, e sicuramente di molto inferiore a 1 su 100).
- Di conseguenza, sono opportune azioni che siano volte a portare l'azienda ad un livello CMM pari a 2.
- Pertanto, possiamo consigliare all'azienda di prendere provvedimenti volti a definire i processi di gestione dei progetti più importanti, in maniera da:
 - Permettere la loro pianificazione
 - Tenere traccia dei costi (tempo e denaro)

Scelta delle azioni (2)

- Ad esempio, possiamo consigliare all'azienda di insistere sulle aree chiave di processo tipiche di CMM 2, ovvero:
 - GESTIONE DEI REQUISITI (RM)
 - PIANIFICAZIONE DEI PROGETTI SW (PP)
 - TRACCIABILITÀ E VISIBILITÀ DEI PROGETTI (PT)
 - GESTIONE DEI SUBAPPALTI (SM)
 - ASSICURAZIONE DELLA QUALITÀ DEL SW (QA)
 - GESTIONE DELLA CONFIGURAZIONE DEL SW (CM)