۱۷

روش‌های آزمایش نرم‌افزار

دکتر عزیزاله محبی گرگری
edu.mandegar@yahoo.com

درس مهندسی نرم‌افزار ۲

مفاهیم کلیدی

- آزمایش نرم‌افزار
- چه کساني نرم‌افزار را آزمایش مي‌کنند؟
- اصول آزمایش
- قابلیت آزمایش
- طراحي داده‌هاي آزمایش (*test case*)
- آزمایش جعبه سفید
- آزمایش جعبه سیاه

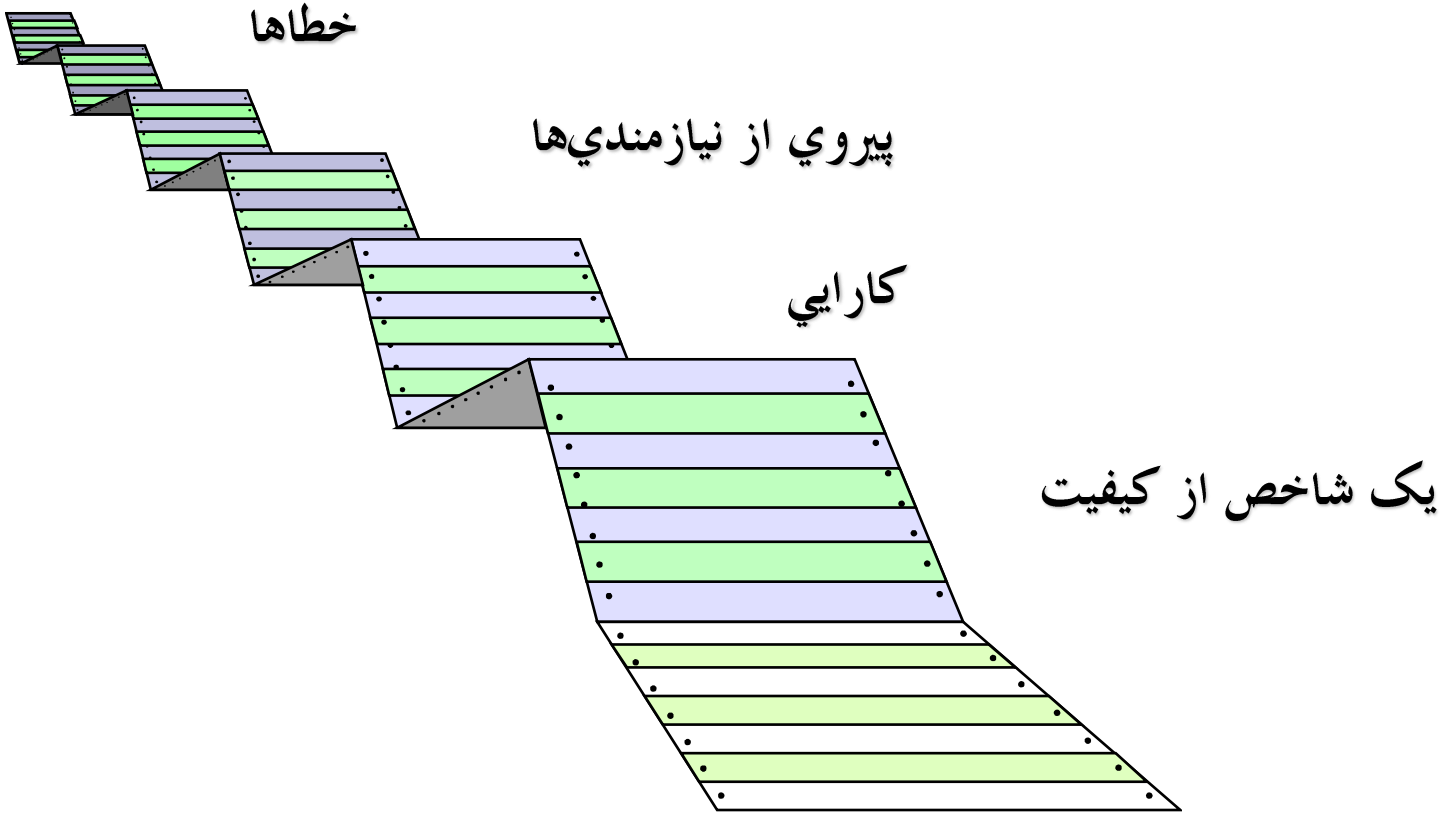
edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

آزمایش نرم‌افزار

- آزمایش نرم‌افزار
  - فرآيند بازبيني نرم‌افزار با هدف يافتن خطاها قبل از تحويل به كاربر نهايي
- اهداف آزمایش نرم‌افزار (*Glen Myers*)
  - آزمایش، فرآيند اجراي برنامه با هدف يافتن خطا است
  - يك داده آزمايش (*test case*) خوب، نمونه‌ای است كه با احتمال بالايي خطاها را بيابد
  - آزمایش موفق، آزمايشي است كه خطاهاي تاكنون يافت نشده را بيابد

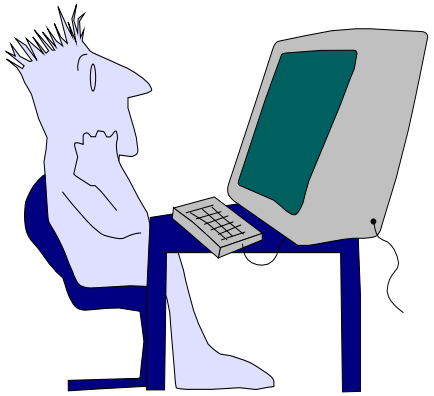
edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

آزمایش نرم‌افزار چه چیزی را نشان می‌دهد؟



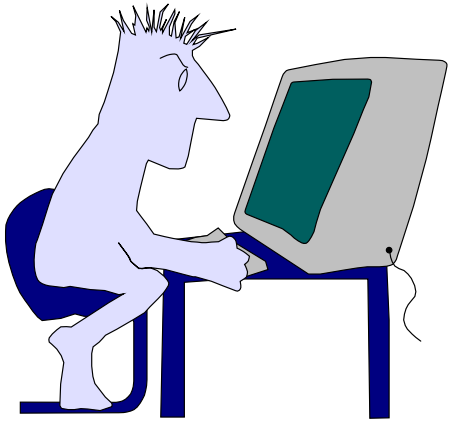
edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

چه کسی نرم‌افزار را آزمایش می‌کند



توسعه دهنده

سیستم را درک مي‌کند،
اما آن را بتدریج آزمایش مي‌کند
و هدفش تحویل نرم‌افزار است



آزمایش‌کننده مستقل

باید عملکرد سیستم را فراگیرد
اما تلاش مي‌کند آن را از کار بیندازد
و هدفش کیفیت است

و در نهایت کاربر نهایي سیستم را آزمایش مي‌کند...

دکتر عزیزاله محبی گرگری
edu.mahdegar@yahoo.com

اصول آزمایش نرم‌افزار (*Davis*)

- تمام آزمایشات باید به نیازمندي‌هاي مشتري قابل ردگيري باشند
- آزمایشات باید مدتي طولاني قبل از شروع، برنامه‌ریزي شوند
- اصل *Pareto* براي آزمایش نرم‌افزار بکار گرفته شود
- آزمایش باید با اجزاء کوچک شروع شود و به سمت آزمایش کل سیستم پیش رود
- آزمایش کامل و جامع امکان‌پذیر نیست
- به منظور تاثیرگذاري بیشتر، آزمایش باید توسط تیم مستقلي هدایت شود

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

قابليت آزمايش

- **عملياتي بودن (*Operability*)**
  - نرم‌افزار هرچه بهتر كاركند، با كارايي بالاتري مي‌تواند آزمايش شود
- **قابليت مشاهده (*Observability*)**
  - آنچه را مي‌توانيد ببينيد، مي‌توانيد آزمايش كنيد
- **قابليت كنترل (*Controllability*)**
  - هر چه نرم‌افزار بيشتر قابل كنترل باشد، آزمايش بيشتر به طرز خودكار و بهينه قابل انجام است
- **پايداري (*Stability*)**
  - هر چه تغييرات كمتر باشد، انحراف از آزمايش كمتر است

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

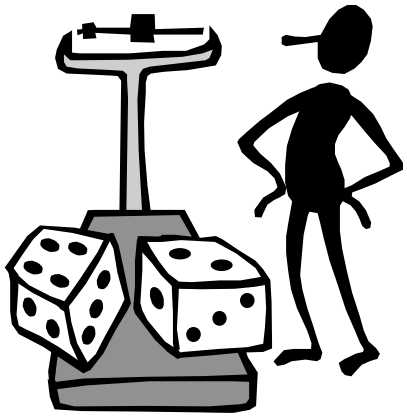
قابلیت آزمایش (ادامه)

- **سادگی (*Simplicity*)**
  - هر چه موارد براي آزمايش كمتر باشد، آزمايش با سرعت بيشتري انجام می‌گيرد
- **قابليت تجزيه‌پذيري (*Decomposability*)**
  - با كنترل نمودن محدودة آزمايش، با سرعت بيشتري مسايل تجزيه مي‌شوند و آزمايشات هوشمندانه‌تري انجام مي‌شود
- **قابليت فهم (*Understandability*)**
  - هر چه اطلاعات بيشتري در اختيار داشته باشد، آزمايش هوشمندانه‌تري انجام مي‌شود

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

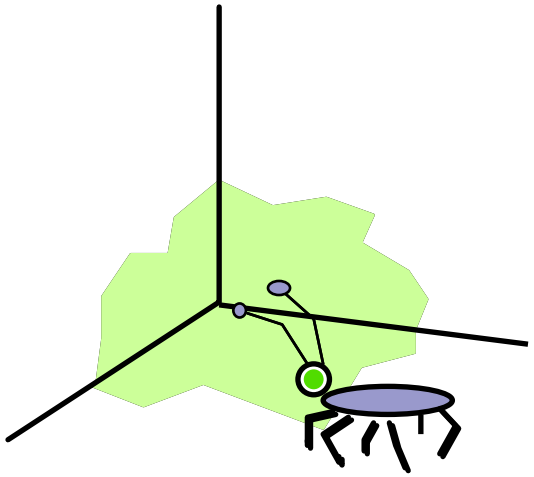
آزمایش خوب

- با احتمال بالا، خطاها را نشان مي‌دهد
- تكراري نيست
- جامعيت خوب و مناسب دارد
- نه بسيار ساده و نه بسيار پيچيده است



edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

طراحی داده‌های آزمایش (*Test case*)



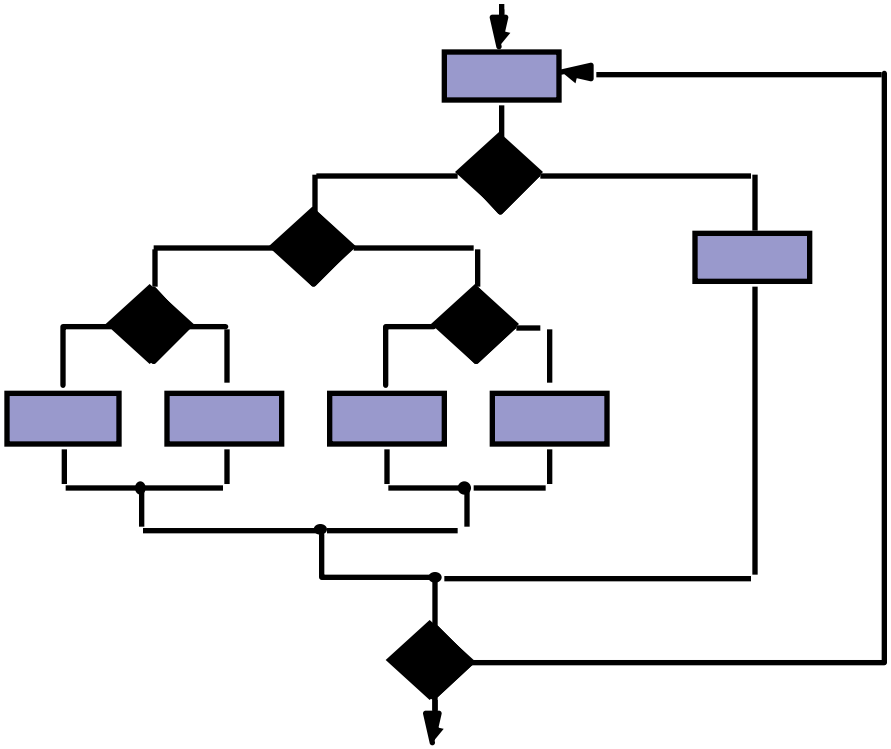
اشکالات در گوشه‌ها کمین مي‌کنند و
در مرزها جمع مي‌شوند

Boris Beizer

- هدف براي نمايش خطاها
- ملاک با روشي کامل
- اجبار با حداقل تلاش و زمان

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

آزمایش کامل

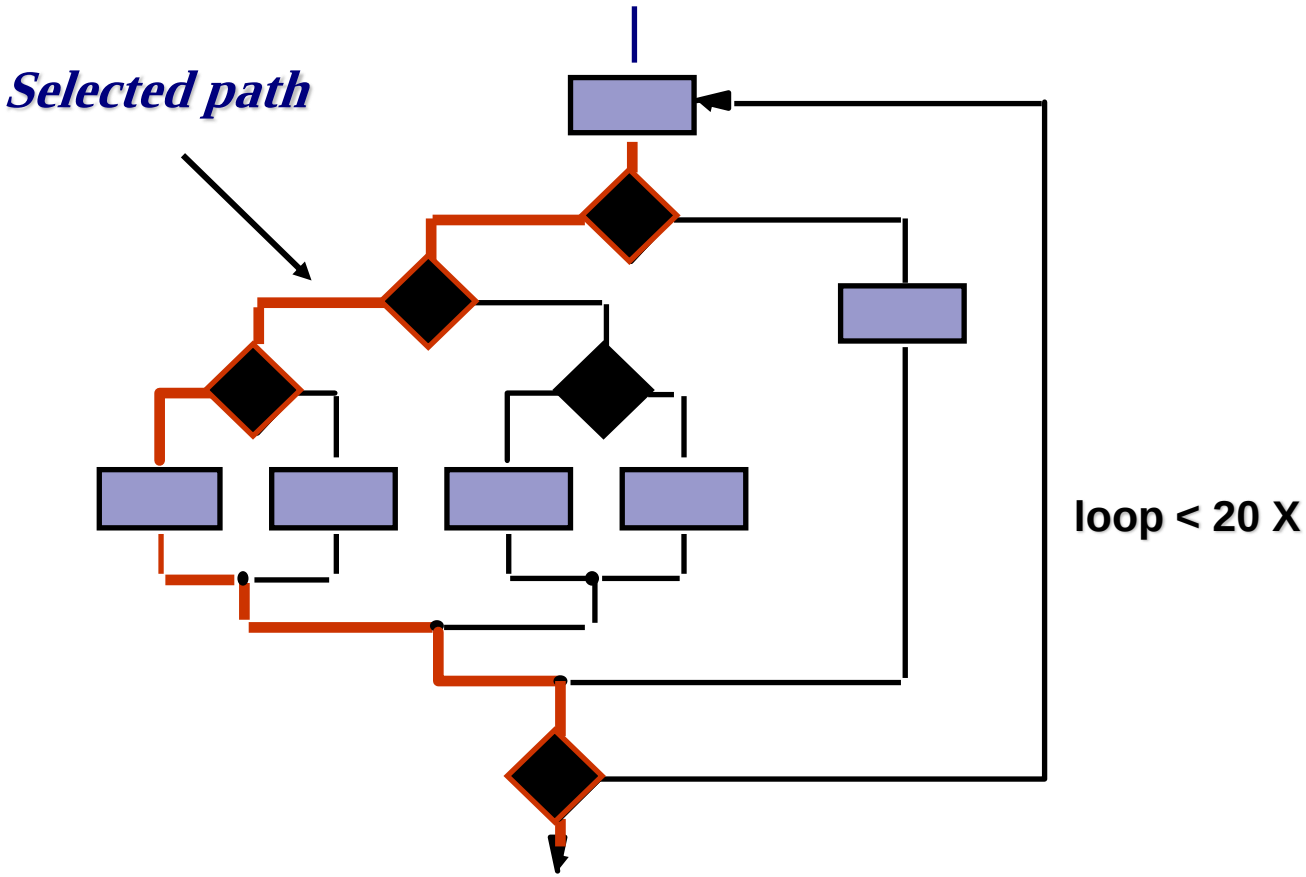


loop < 20 X

- براي آزمايش كامل يک برنامه ۱۰۰ خطي داراي دو حلقه تودرتو و چهار شرط كه يک تا ۲۰ بار اجرا مي‌شوند تقريباً به $۱۰^{۱۴}$ مسير داريم كه آزمايش كامل آنها ۳۱۷۰ سال طول مي‌كشد!

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

آزمایش انتخابی



edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

روش‌های آزمایش انتخابی نرم‌افزار

- **جعبه سیاه**
  - با دانستن تابع خاصي كه يك محصول براي انجام آن طراحي شده
- **جعبه سفید**
  - با دانستن عملكرد داخلي محصول

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

استخراج مسیرهای پایه

- مقدار محاسبه شده براي پيچيدگي دوراني، تعداد مسيرهاي مستقل را در مجموعة پايه برنامه مشخص مي كند
  - حد بالايي براي تعداد آزمايشاتي كه بايد براي اطمينان از اجراي حداقل يك بار هر يك از دستورات انجام شوند
  - مسير مستقل؛ هر مسيري در برنامه كه حداقل يك مجموعة جديد از دستورات پردازشي يا شرط جديدي را بيان نمايد
- پس از استخراج مسيرهاي پايه، داده آزمايش براي هر مسير ايجاد مي‌شود

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه از استخراج مسیرهای پایه

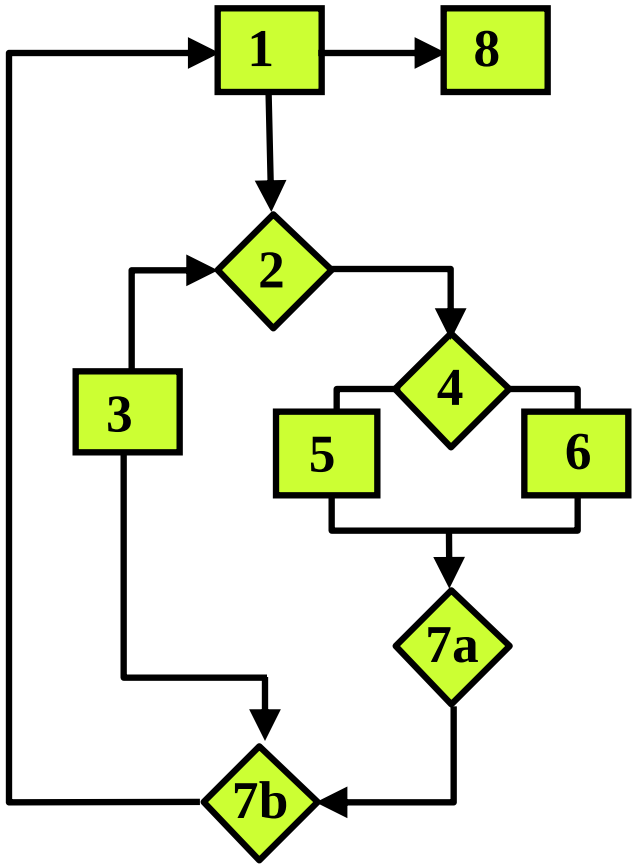
Example PDL

```
        procedure  sort
1 :     do while records remain
                read record ;
2 :             if  record field1 = 0
3 :                     then  process record ;
                                store in buffer ;
                                 increment counter ;
4 :                 elseif  record field2 = 0
5 :                         then reset counter ;
6 :                         else  process  record ;
                                  store in file ;
7a:                         endif
                    endif
7b:             enddo
8 :end
```


edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه از استخراج مسیرهای پایه (ادامه)

- مقدار پیچیدگي دوراني



$$V(G) = 11 - 9 + 2 = 3 + 1 = 4$$

Path 1: 1,8

Path 2: 1,2,3,7b,1,8

Path 3: 1,2,4,6,7a,7b,1,8

Path 4: 1,2,4,5,7a,7b,1,8

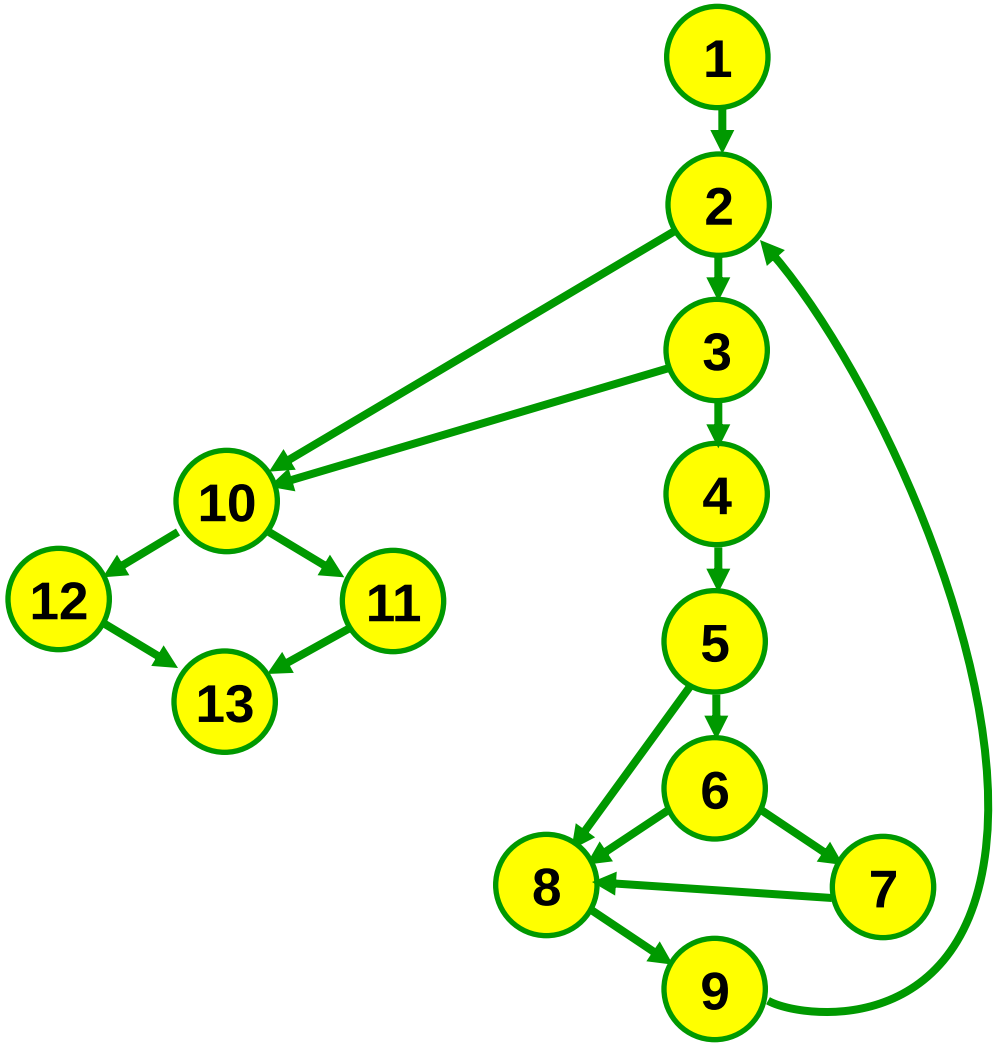
edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه دیگر از استخراج مسیرهای پایه

```
PROCEDURE average;

    *  This procedure computes the average of 100 or fewer
       numbers that lie between bounding values; it also computes the
       sum and the total number valid.

    INTERFACE RETURNS average, total.input, total.valid;
    INTERFACE ACCEPTS value, minimum, maximum;

    TYPE value[1:100] IS SCALAR ARRAY;
    TYPE average, total.input, total.valid;
       minimum, maximum, sum IS SCALAR;
    TYPE i IS INTEGER;
    i = 1;
    total.input = total.valid = 0;
    sum = 0;
    DO WHILE value[i] <> -999 AND total.input < 100
       increment total.input by 1;
       IF value[i] > = minimum AND value[i] < = maximum
          THEN increment total.valid by 1;
                  sum = s sum + value[i]
          ELSE skip
       ENDIF
       increment i by 1;
    ENDDO
    IF total.valid > 0
       THEN average = sum / total.valid;
       ELSE average = -999;
    ENDIF
END average
```


edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه دیگر از استخراج مسیرهای پایه (ادامه)



edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه دیگر از استخراج مسیرهای پایه (ادامه)

V(G) = 6 regions , V(G) = 17 edges – 13 nodes + 2 = 6 ,

V(G) = 5 predicates nodes + 1 = 6

path 1: 1-2-10-11-13

path 2: 1-2-10-12-13

path 3: 1-2-3-10-11-13

path 4: 1-2-3-4-5-8-9-2-. . .

path 5: 1-2-3-4-5-6-8-9-2-. . .

path 6: 1-2-3-4-5-6-7-8-9-2-. . .

edu.mandegar@yahoo.com دکتر عزیزاله محبی گرگری

نمونه دیگر از استخراج مسیرهای پایه (ادامه)

Path 1 test case:

value(k) = valid input, where $k < i$ for $2 \le i \le 100$

value(i) = -999 where $2 \le i \le 100$

Expected results: Correct average based on k values and proper totals.

Note: Path 1 cannot be tested stand-alone but must be tested as part of path 4, 5, and 6 tests.

Path 2 test case:

value(1) = -999

Expected results: Average = -999; other totals at initial values.

Path 3 test case:

Attempt to process 101 or more values.

First 100 values should be valid.

Expected results: Same as test case 1.

Path 4 test case:

value(i) = valid input where i < 100

value(k) < minimum where $k < i$

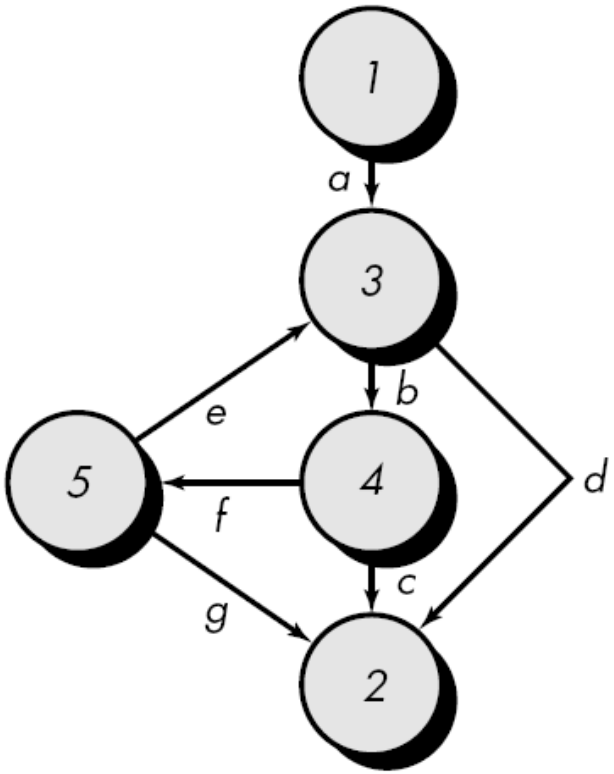
Expected results: Correct average based on k values and proper totals.

ماتریس گراف (*Graph Metric*)

- ماتريس مربعي است كه اندازة آن (يعني تعداد سطرها و ستون‌ها) برابر است با تعداد گره‌ها در گراف جريان
- هر سطر و ستون معادل يك گرة مشخص است و ورودي‌هاي ماتريس معادل يال‌هاي بين گره‌ها مي‌باشد
- با افزودن وزن يال به هر ورودي ماتريس، ماتريس گراف به ابزاري قدرتمند براي ارزيابي ساختار كنترل برنامه در ضمن آزمايش تبديل شود

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه ماتریس گراف



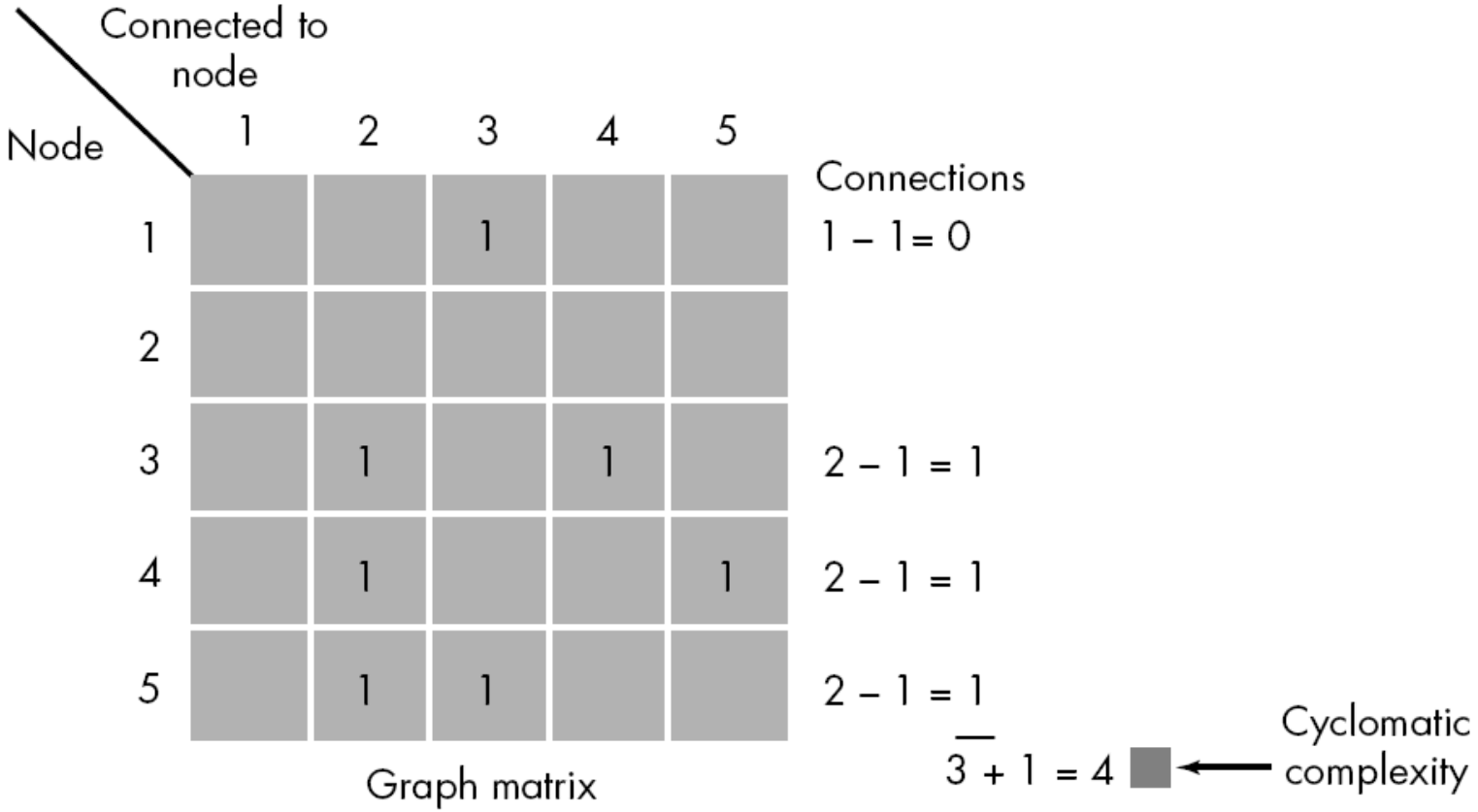
Flow graph

Node \ Connected to node	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

Graph matrix

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

نمونه ماتریس گراف (ادامه)



edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری

پایان

edu.mandegar@yahoo.com
دکتر عزیزاله محبی گرگری