Programación Avanzada

Prof.Pablo Castro
Depto. Computación
UNRC

Horarios

- **Teóricos** (Prof.Pablo Castro)
  - Miércoles: 14-16hs (Aula 31 Pab. 4)
  - Viernes: 10-12hs (Aula 34 Pab. 4)

- **Laboratorios** (Prof.Marta Novaira)
  - Martes: 10-12hs (Aula 101)
  - Lunes: 14-16hs (Aula 101)
- **Prácticos** (Prof. Simon Gutierrez, Prof. Ernesto Cerdá, Luciano Putruele)
  - Martes 16-18hs (Aula 101)
  - Jueves 8-10hs (Aula 101)

Contactos

- Prof. Pablo Castro (pcastro@dc.exa.unrc.edu.ar)
- Prof. Ernesto Cerdá (ecerda@dc.exa.unrc.edu.ar)
- Prof. Marta Novaira (mnovaira@dc.exa.unrc.edu.ar)
- Prof. Simon Gutierrez (sgutierrez@dc.exa.unrc.edu.ar)
- Prof. Luciano Putruele (lputruele@dc.exa.unrc.edu.ar)

La página de la materia puede acceder por medio de:

Google classroom: classroom.google.com

Para registrarse:

Código: 6a4ylko

Modalidad de la Materia

- Para regularizar la materia:
  - Aprobar dos parciales con nota mayor o igual a 5,
  - Aprobar los trabajos prácticos

Para aprobar la materia:

- Aprobar el examen final (escrito u oral).

Temas de la Materia

En la materia nos enfocaremos en tratar de construir programas **correctos**:

Un programa es correcto si hace exactamente lo que es requerido por su especificación

Temas Específicos

- Repaso de lógica,
- Programación funcional y calculo de programas funcionales,
- Lógica de Hoare y Corrección de Programas Imperativos,
- Nociones Básicas de Automátas y Lenguajes,
- Nociones Básicas de Complejidad.

Bibliografía

- **Cálculo de Programas**, *Autores: Javier Blanco, Silvina Smith, Damián Barsotti*. Está disponible en .pdf

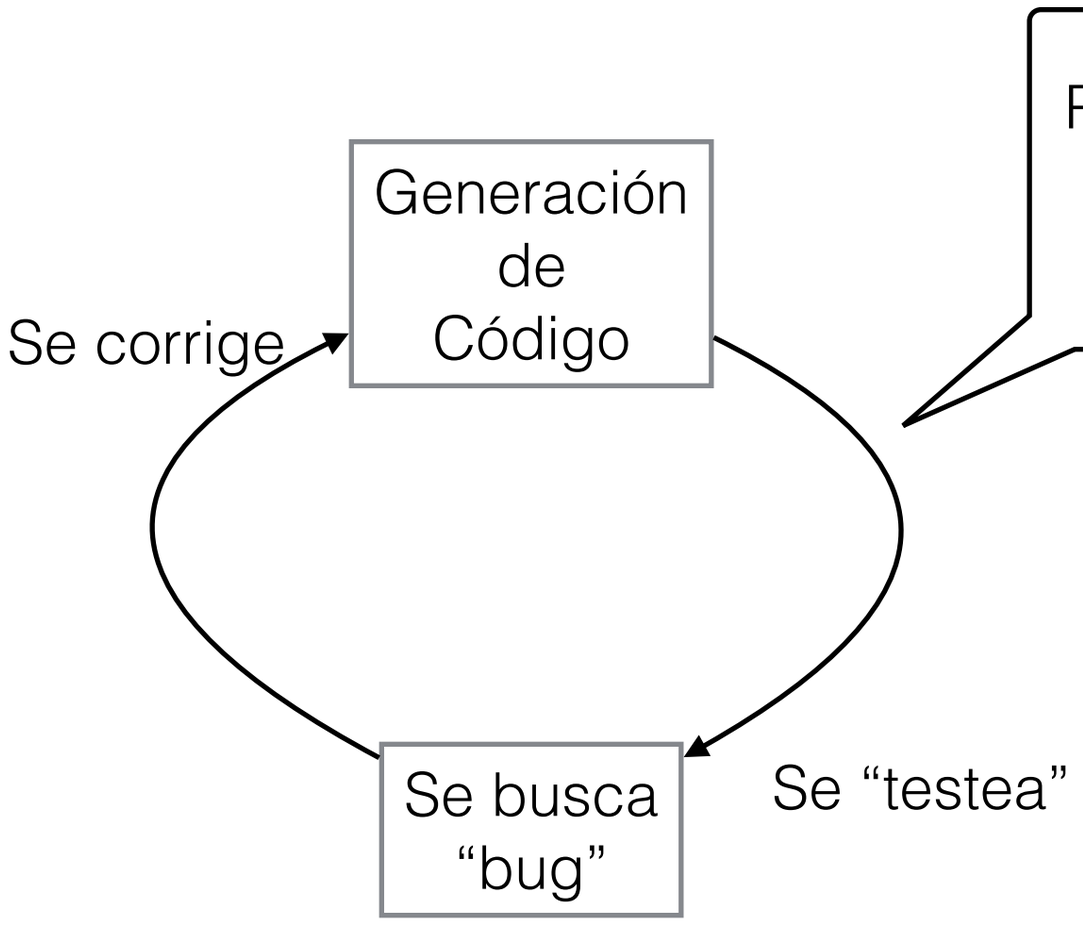
- **Program Construction, Calculating Programs from Implementations**, *Autores: Rolland Backhouse.* Disponible en la biblioteca.

- **Introduction to Functional Programming using Haskell**. *Autor: Richard Bird*. Disponible en la biblioteca

- **Learn you a Haskell for Great Good.** Disponible online ("Aprende Haskell por el bien de todos" en español)

Programación por Prueba-Error

Una forma de programar es por prueba y error:



"El testing demuestra la presencia de errores pero no su ausencia"
E.Dijkstra

¿Cuán Peligrosos son los Bugs?

En aplicaciones críticas los bugs pueden ser muy peligrosos:

- El caso del Ariane 5:

Un error de overflow hizo que el Ariane calcule mal las trayectorias y se destruya

Más Bugs

- La Therac 25

Una máquina para radioterapia, mató a varios pacientes debido a un error de software.

Más Bugs

- Airbags

General Motors retiró del mercado más de un millon de autos (96-97) por errores del software que controlaba los airbags (chevrolet cavaliers)

Más Recientemente…

Dos aviones Boeing 737-MAX se estrellaron por posibles fallas de sus sistemas de vuelo

Una falla de un sensor en los aviones hace que el sistema de vuelo reaccione de forma peligrosa, incluso sin tener en cuenta las ordenes del piloto.

Ejemplo de Razonamiento Riguroso

Tenemos una barra de chocolate y queremos saber cuantos cortes tenemos que hacer para quedarnos con todos los cortes

¿Cómo solucionamos este problema?

Usando la lógica y la matemática

Prueba y Error…

Definamos:

$$B: \text{ cantidad de bloques totales}$$

$$C: \text{ cantidad de cortes realizados}$$

$$P: \text{ cantidad de partes obtenidas hasta el momento}$$

Problema del Chocolate

Si pensamos un rato encontramos el siguiente invariante:

$$P - C = 1$$

Esta propiedad se llama invariante debido a que vale siempre durante el proceso

Queremos llegar a: $P = B$

Cantidad de partes igual a cantidad de bloques

Veamos:

$$P - C = 1 \equiv B - C = 1 \equiv C = B - 1$$

Reemplazando

Despejando

Cantidad de cortes necesarios

El Problema de la Torta

Consideremos una torta:

- Tenemos N puntos en el contorno
- Se trazan todas la cuerdas posibles entre los puntos
- Nunca se cortan mas de dos cuerdas en cada punto de intersección
- ¿Cuántas porciones obtenemos?

Prueba y Error

Podemos intentar por prueba y error:

1 punto

1 porción

2 puntos

2 porciones

3 puntos

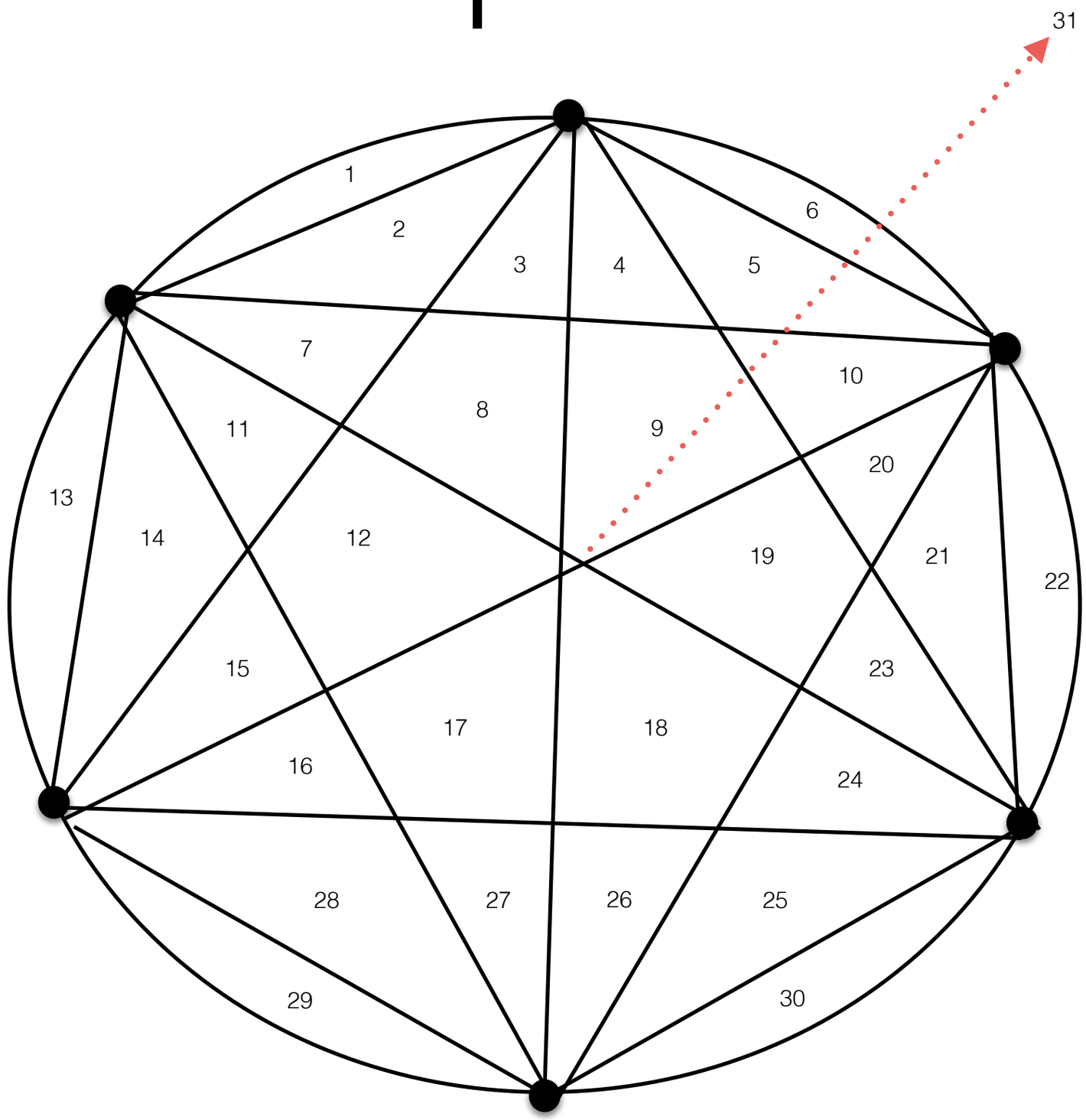
4 porciones

4 puntos

8 porciones

Podríamos pensar que para n, tenemos 2^{n-1}porciones

Pero para 6


31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Problema de la Torta

Sin embargo… Para 6 puntos son 31 porciones

Tratemos de razonar rigurosamente:

f : número de porciones

c : número de cuerdas

p : cantidad de intersecciones internas

Tratemos de razonar rigurosamente:

Problema de la Torta

Analicemos el problema:

número de porciones agregadas por una cuerda

$=$ { las cuerdas dividen porciones en dos }

número de porciones cortadas por la cuerda

$=$ {Una porción se divide por un segmento de la cuerda}

número de segmentos de la cuerda

$=$ {Los segmentos estan determinados por la cantidad de puntos de intersección}

$1 +$ cantidad de puntos de intersección

Es decir:

Cantidad de porciones agregadas por c cuerdas

$=$ { deducción anterior }

c $+$ Cantidad de puntos de intersección en las c cuerdas

Problema de la Torta

Es decir:

Al comienzo tenemos 1 porción

$$f = 1 + c + p$$

Podemos calcular:

$$\begin{aligned}
& f \\
&= \{ \text{ formula anterior } \} \\
& 1 + c + p \\
&= \{\text{una cuerda cada dos puntos}\} \\
& 1 + \binom{N}{2} + p \\
&= \{\text{un punto de intersección cada cuatro puntos}\} \\
& 1 + \binom{N}{2} + \binom{N}{4} \\
&= \{\mathit{Algebra}\} \\
& 1 + \frac{N^4 - 6N^3 + 23N^2 - 18N}{24}
\end{aligned}$$

Difícil de obtener con prueba y error!