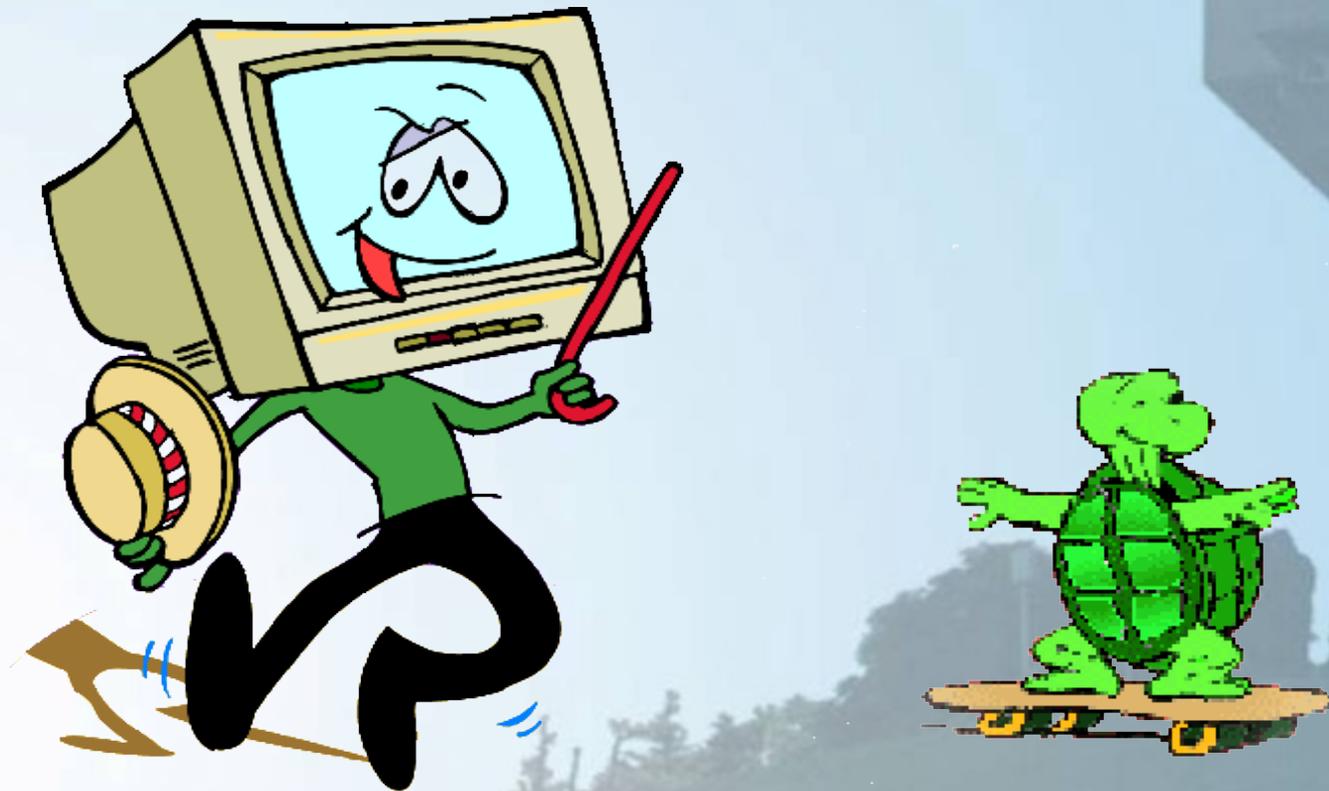


# A la compu... ¿se le escapa la tortuga?

**Dra. Isabel Méndez-Díaz**  
Departamento de Computación  
FCEyN-UBA



# ¿Qué es un Problema Combinatorio?

- Es un problema en que deben contarse una cierta cantidad de posibilidades, casos, configuraciones, conjuntos, etc.
- El resultado es un número entero.



## Ejemplo de Problema Combinatorio

¿De cuántas formas diferentes pueden sentarse  $n$  alumnos en un aula de  $n$  asientos?

Respuesta:

Las podemos recorrer de

$$n! = 1 \times 2 \times 3 \times 4 \times 5 \dots (n-2) \times (n-1) \times n$$

maneras distintas





400!

64034522846623895262347970319503005850702583026002  
95945868444594280239716918683143627847864746326467  
62943505750358568108482981628835174352289619886468  
02997937341654150838162426461942352307046244325015  
11444867089066277391491811733195599644070954967134  
52904770203224349112107975932807951015453726672516  
27877890009349763765710326350331533965349868386831  
33935202437378815778679150631185870261827016981974  
00629830253085912983461622723045583395207596115053  
02236086810433297255194852674432232438669948422404  
23259980555161063594237696139923191713406385899653  
79701478272066063202173794720103213566246138090779  
42304597360699567595836096158715129913822286578579  
54936161765448045322200782581840084843641559122945  
42753848035583745180226759000613995601455952061272  
1119291810503249100800000000000000000000000000000000  
000  
000



# ¿Qué es un Problema de Optimización Combinatoria?

- Es un problema en el cual, de un conjunto de objetos valuados con alguna función, se busca el objeto con “mejor” valor.
- Puede ser el objeto de máximo valor, de mínimo valor o alguna otra alternativa.



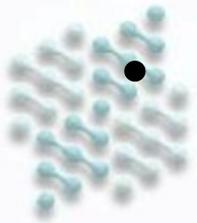
# ¿Para qué Sirven?

Modelizan algunos problemas de la realidad que tienen importantes aplicaciones prácticas.



# Ejemplos de Problemas

- Asignación de tareas.
- Localización.
- Cortes de materia prima.
- Venta de pasajes aéreos.
- Asignación de tripulación.
- Horarios.
- Fixture de un campeonato
- Distribución en contenedores



# ¿Cómo se Resuelve un Problema de Optimización Combinatoria?

## Fuerza Bruta

Consiste en listar todos los casos y para cada uno calcular su costo, identificando de este modo el caso de costo más conveniente.

Hoy contamos con las computadoras, capaces de hacer millones de operaciones aritméticas y lógicas por segundo.

Podríamos pensar que como son muy eficientes y rápidas no tendremos problemas en resolver los problemas más grandes que se nos presenten.

¿Será cierto?



# Problema del Viajante

Un viajante debe recorrer cierta cantidad de ciudades y volver finalmente a la ciudad donde vive.

¿Cuál es el mejor recorrido?

- ✓ El más corto.
- ✓ El más rápido.



# Fuerza Bruta

Supongamos que quiero resolver el problema del viajante de comercio para 20 ciudades

¿Cuánto creen que tardaremos en evaluar todos los posibles recorridos?

¡Arriesguen!

1 minuto, 1 hora, 1 día, 1 año, 1 siglo o más?



# Fuerza Bruta

Tenemos una computadora que realiza un billón de evaluaciones por segundo  
(1.000.000.000.000 x seg)

Hay que evaluar  
**20!=2432902008176640000** posibilidades

entonces tardaríamos...

2.432.902 seg

= 675 horas =28 dias



# Fuerza Bruta

Tenemos una computadora que realiza un billón de evaluaciones por segundo (1.000.000.000.000 x seg)

Hay que evaluar

**30!=265252859812191058636308480000000**

posibilidades

entonces tardaríamos ...

73.681.349.947.830.849 horas

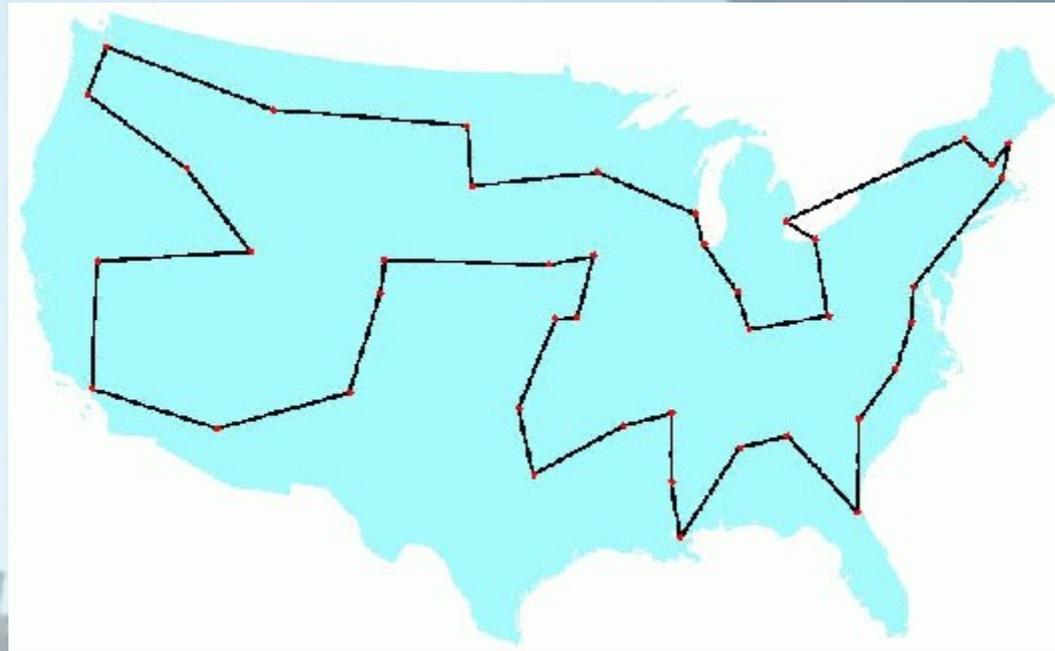
3.070.056.247.826.285 días = 8.411.113.007.743 años

**8.411.113.007 siglos**

# Más Allá de la Fuerza Bruta

En 1954 Dantzig, Fulkerson y Johnson resolvieron un caso de 49 ciudades.

DF&J estaban seguros de que la solución era la mejor del conjunto de  $49!$  soluciones posibles.





# Solución Record de 15.112 Ciudades (2001)

- Resuelta en una red de 110 máquinas en las universidades de Rice y Princeton.
- Tiempo total de cómputo de 22.6 años de una Compaq EV6 Alpha de 500 MHz
- Longitud total de aproximadamente 66.000 Km (Un poco más de una vuelta y media a la tierra por el ecuador).





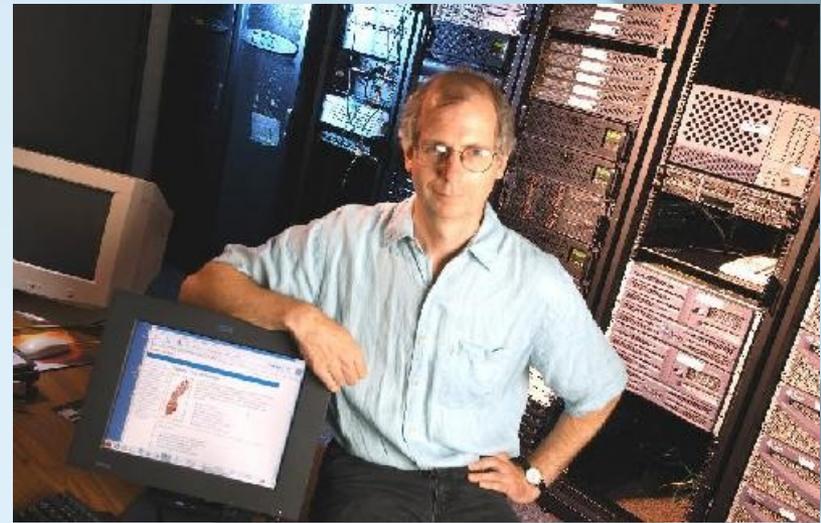
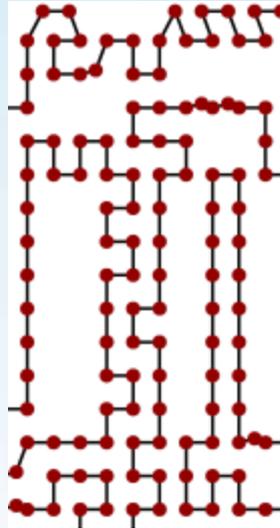
# ¿Como evolucionó la búsqueda de la solución?

- 855618 km 2001
- 855612 km 2001
- 855610 km 2001
- 855602 km 2003
- 855597 km 2003
- 855597 km 2004



# Solución record 85900 ciudades

2006



96 Nodos, 2.8 GHz y 2 GBytes.



# ¿Cómo evolucionó la búsqueda de la solución?

Fecha	Diferencia	Dias de CPU
Dic.2004	0.0209%	269.5
Marzo 2005	0.0178%	187.8
Mayo 2005	0.0150%	3539.0
Julio 2005	0.0060%	5160.6
Julio 2005	0.0073%	750.9
Julio.2005	0.0054%	1499.1
Agos 2005	0.0036%	7702.7
Agos.2005	0.0049%	347.1
Sept 2005	0.0044%	527.2
Oct 2005	0.0030%	1996.0
Nov 2005	0.0027%	4305.7
Nov 2005	0.0022%	1181.6
Nov.2005	0.0016%	1961.1
Dic 2005	0.0019%	1396.1
Dic 2005	0.0016%	1317.0
Marzo 2006	0.0007%	7748.9
Marzo 2006	Optimo	3698.0
Abril 2006	Optimo	2719.5
Abril 2006	Optimo	512.9
Abril 2006	Optimo	286.2



# Problema abierto: 1.904.711 ciudades



H. Nguyen, I. Yoshihara, K. Yamamori y M. Yasunaga  
7.518.425.642 Junio 2003

Keld Helsgaun

7.517.285.610m	Septiembre 2003
7.515.971.188m	Mayo 2007
7.515.947.511m	Noviembre 2008
<b>7.515.877.991m</b>	Mayo 2009

Valor mínimo **7.512.218.268m** Junio 2009 **Error: 0.0487%**





# ¿Jugamos al prode?

Tenemos  $n$  partidos.

Debemos decir si el resultado es local, empate o visitante.

Gano premio si me equivoco en a lo sumo un partido.

¿Cuál es la cantidad mínima de boletas que debo jugar para asegurarme ganar?



## Boletas con 1 partido

	Local	Visitante	Empate
Boca-River	X		

Cualquiera sea el resultado, es boleto ganadora



## Boletas con 2 partidos

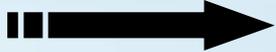
	Local	Visitante	Empate
Boca-River	X		
Vélez-Racing	X		

	Local	Visitante	Empate
Boca-River		X	
Vélez-Racing	X		

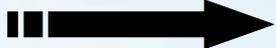
	Local	Visitante	Empate
Boca-River			X
Vélez-Racing	X		



# Resultados conocidos

1        1

2        3

3        5

4        9

5        27

 6         $65 \leq ? \leq 73$  (hace más de 40 años)

Jeff Linderth (University of Wisconsin, Abril 2006)

Tiempo	47.15 días
Tiempo de CPU	57.31 años
Promedio de computadoras	467.3
Máximo de computadoras	1253

Para demostrar que  $70 \leq 65 \leq ? \leq 73$





**Error en el código...**



Jeff Linderoth (University of Wisconsin, Septiembre 2006)

Tiempo	72.3 días		
Tiempo de CPU	110.01 años		
Promedio de computadoras	555.8		
Máximo de computadoras	2038	70	<del>65</del> $\leq ? \leq 73$



Jeff Linderoth (University of Wisconsin, Mayo 2007)

Tiempo	19.7 días		
Tiempo de CPU	30.3 años		
Promedio de computadoras	562.4		
Máximo de computadoras	1775	71 <del>70</del> <= ? <= 73	



# Problemas Fáciles y Difíciles

- ✓ **Problema fácil:** Verificar que recorro todas las ciudades.
- ✓ **Problema difícil :** Verificar que es el camino mas corto.



# Resolución Satisfactoria

¿Cuándo se considera que la computadora resuelve satisfactoriamente un problema?

Cuando el tiempo que tarda en encontrar la solución es “razonable”.



Hay problemas para los cuales las computadoras encuentran la solución en tiempo “razonable”.

Hay problemas para los cuales **TODAVÍA** las computadoras no han podido encontrar la solución en tiempo “razonable”.



¿POR QUÉ?

NO LO SABEMOS...

¿CARACTERÍSTICAS DEL PROBLEMA?

¿LIMITACIÓN HUMANA?

EL DESAFÍO ESTÁ PLANTEADO...

