

818181181818

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu Jessica Meza-Jaque – je.meza@uandresbello.edu

OBJETIVOS DE LA SESIÓN

- Conocer los conceptos de clases de problema P y NP
- Relacionar estas clases en cuanto a sus características
- Analizar ejemplos de problemas clasificados en estas clases



CONTENIDOS DE LA SESIÓN



- Concepto de clase P
- Concepto de clase
 NP
- Muestra de ejemplos

Recordemos

- Problemas de Búsqueda
- Problemas de Decisión

Problemas de Búsqueda

- Dada una instancia encontrar una solución correspondiente.
- También se puede determinar que no existe solución o ésta no está definida (devolver λ o 1).
- Corresponde a la noción de "Resolver un problema" en la vida real.

Problemas de Decisión

- Dada una instancia, determinar si la instancia se encuentra en un grupo especificado (la instancia cumple una condición).
- La solución puede ser 'si' o 'no'

P versus NP

 Cotidianamente nuestra experiencia nos muestra que resolver problemas por lo general es más difícil que determinar si su solución es correcta.

 ¿Es esto coincidencia o es la representación de un hecho real en la vida?

 De esta manera podemos ver informalmente es sentido de la pregunta P vs NP

Clase de problemas P

- La clase de **problemas P** representa los **Problemas de decisión** que pueden ser **Eficientemente** resueltos.
- Definimos como **Eficiente** como una solución que puede ser expresada como un polinomio en función del **Tamaño de su entrada** (importante relación para la exposición presentada desde ahora).

P es por Polynomial-time

Clase de problemas P - Ejemplos

- Ordenar un conjunto de números
 - Solución O(n*logn) usando el algoritmo de ordenamiento Quicksort
- Encontrar un número en un arreglo
 - Solución O(n) usando búsqueda secuencial
 - Solución O(logn) usando búsqueda binaria
- Encontrar la ruta más corta entre 2 nodos de un grafo
 - Solución O(V + E) Usando el algoritmo Depth-First Search (Búsqueda en profundidad)

Clase de Problemas NP

- La clase de Problemas NP representa los problemas que pueden ser resueltos en tiempo polinómico en una máquina de Turing (MT) nodeterminista.
- También podemos definirlo como el grupo de problemas de decisión cuya solución puede ser verificada en tiempo polinómico.

NP es por Non-deterministic Polynomial-time

Clase de Problemas NP

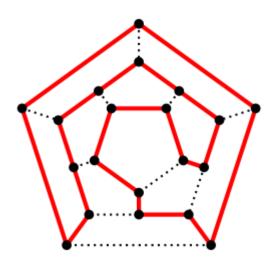
- Una MT no determinista es una máquina que utilizando una misma entrada podría entregar muchos posibles resultados sin poder determinarse el "camino" para lograrlo.
- Podemos llamarlo también Algoritmo No determinista

Clase de Problemas NP - Ejemplos

- Todos los problemas en P pertenecen a NP
 - Una MT no-determinista puede funcionar como una MT determinista.
 - Podemos determinar que cualquier solución de un problema P es válida.

Clase de problemas NP - ejemplo

- Determinar el ciclo hamiltoniano en un grafo, esto implica encontrar una ruta que pase a través de todos los nodos del grafo y vuelva al inicio.
 - Se puede resolver con heurísticas.
 - La solución es fácilmente verificable O(n) si podemos identificar que todos los nodos están marcados en la ruta elegida.



Intratabilidad en la computación

- Decimos que un problema es intratable cuando:
 - Solucionarlo requiere más que un tiempo polinomial en función del tamaño de su entrada.
 - Problemas que no tienen una solución (asumiendo hasta el día de hoy que existen).

• Ejemplos:

- Problemas cuya única solución encontrada hasta ahora es de tiempo exponencial (solo pueden ser resueltos por fuerza bruta).
- Halting Problem (No tiene solución).

Formalicemos la definición de P y NP

 Podemos definir las clases de problemas P y NP en términos de problemas de Búsqueda y Problemas de decisión.

En función de "Problemas de búsqueda"

La clase P como un problema de búsqueda natural

• Decimos que P es la agrupación de problemas de búsqueda cuya solución pude ser encontrada (cuando existe) eficientemente.

- R \underline{c} {0, 1}* x {0, 1}* se puede resolver eficientemente si:
 - Existe un algoritmo en tiempo polinómico A dado que para cada entrada x tenemos que:
 - Si R(x) = {y : (x, y) ∈ R} no es vacío, entonces A(x) resuelve el problema de búsqueda R
 - Y de otra forma, A(x) = 1 (indicando que x no tiene solución)

La clase P como un problema de búsqueda natural

- Detonaremos por *PF (Polynomial-time Find)* la clase de problemas de búsqueda que son eficientemente resolubles.
- Esto es:
 - R ∈ PF si R tiene una solución que podemos encontrar eficientemente.
 - Y existe un algoritmo que en tiempo polinómico resuelve el problema de R.

La clase P como un problema de búsqueda natural

Otras consideraciones:

- Si para la instancia x, no existe una solución (por ejemplo $R(x) = \emptyset$) entonces claramente el algoritmo **A** no resuelve $R(A(x) \notin R(x))$.
- En este caso se requiere que $A(x) = \bot$.
- Por lo tanto A **SIEMPRE** entrega una respuesta correcta, en este caso indicando que x no tiene una solución.

En función de "Problemas de búsqueda"

La clase NP como un problema de búsqueda natural

• Los problemas de búsqueda naturales tienen la propiedad que las soluciones válidas (para ellos) pueden ser **eficientemente** reconocibles.

• Esto es:

- Dada una instancia x para el problema R y una solución candidata y, se puede identificar cuando y es una solución válida o no para x con respecto al problema R.
- $y \in R(x) \circ y \notin R(x)$

La clase NP como un problema de búsqueda natural

- El problema de búsqueda **R**, dado que:
 - R \underline{c} {0, 1}* x {0, 1}*
- Tiene una solución verificable si existe un algoritmo A, dado que:
 - Para cada $x \in y$, A(x, y) = 1 si y solo si $(x, y) \in R$.
 - A(x, y) = 0 de lo contrario.
 - En palabras simples: A comprueba si y es una solución válida para x en R

La clase NP como un problema de búsqueda natural

• Denotaremos por *PC* (Polynomial-time Check) a la clase de problemas de búsqueda cuya solución puede ser **verificable** eficientemente.

 La clase PC es el dominio para el estudio de cuales problemas están en PF ya que la habilidad para reconocer una solución valida es un prerrequisito para discutir acerca de la complejidad de encontrar una solución.

• ¿Qué pasaría si cada problea en PC estuviera también en PF?

 Si PC <u>c</u> PF se consideraría que cuando la solución de las instancias pueden ser eficientemente verificables (que sean correctas), también es el caso de que esas soluciones pueden ser encontradas eficientemente.

• En palabras simples, todos los problemas razonables en PC son fáciles de resolver.

• Por otra parte, si PC \ PF no es vacío, entonces existen problemas razonables que son difíciles de resolver.

• Esto conforma la intuición básica de que algunos problemas son fáciles de resolver mientras que otros son difíciles de resolver.

• Ejemplo:

• Considere varios juegos de puzzle, como rompecabezas, laberintos, sudoku, etc.



Rompecabezas 4 piezas

Rompecabezas 1000 piezas



9	6	3	1	7	4	2	5	8
1	7	8	3	2	5	6	4	9
2	5	4	6	8	9	7	3	1
8	2	1	4	3	7	5	9	6
4	9	6	8	5	2	3	1	7
7	3	5	9	6	1	8	2	4
5	8	9	7	1	3	4	6	2
3	1	7	2	4	6	9	8	5
6	4	2	5	9	8	1	7	3

Sudoku 3x3

Sudoku 6x6

33		17/2	10	15	10	9	2	3	0	24		0			25	20	22	.11	50		4.7			23	37	7		4	34	3		19	30	34	
Ŷ		3	22	*	.0					31		3		28		No.	17	2		30	19	29	35	19	32	20	15	4	11	14	.8			33	
•		21	18			1		6		29		5			0	26			4	28	15	34	32			25	31	30	38	13				35	Г
8		21	3			28	4	10	30			18	20			36	31	2.5	25			17		24	21	8		di l	100	10		9	3		Г
8	30	100	17	36	13	10	35	1			12	34		7	4		27					21		28	-	14	2	5	22	21		15		11	1
90	35	11		5	12		22	17	13	25	9		30		3.	250	18	23	-	33		31	7	19	34		6	26	27	4	24	29	28	4	3
11	7		32	28		15	36	5/	12	10	29	21		4	35	8	1	10	14	24	1,6	20	2	5	100	13	17	0	34	18	(6)	33	25	26	
			15		9		8	1	32	16	3				0			7	34	9	14		-		35	30	18	23		28	3		21:		Ī
18	20		8		29	24		*	1	-22	30	9		25	7	17		1108	0	52		2						80	19	16	4	35		35	Ī
20		30	10	23	22	17				21		18	14	5		19	20	13			29		0		2	11	7		32				4		Г
35	17	0		54	3	2		9	1		5	24	34	16	12			18	31	36	8	Jis	28	25		26	1	10	9		23	6	32		3
1			1	12	4	18	7			14	U		31		8			17	40		25	30		33	20			21	10	19	11	10	34	24	Ē
28		30	35		47	10		13			18	36		20		22	21	14		3	1	27	16		9	8			55	26	1		24	2	ſ
		7			-	27	21	23		3			17			33		15	20				22	35	24				ø			11	31	4	9
22		4	5		31		32	24	8	1	13	1	25	19	26	29		28	2	21	- 13	18		10	33	34		1	7	35	27	13		17	. 5
16	8	14	9	32	95	35		19	2			13	27		24			34	17	31	U.	1			3			18	26	20		0	22	30	7
24	110	13	25	18	26	7					31	6	12	35		100	16	36		32	5	33			11	17		2				3	29	23	3
7		23	3	27	X					17	29	5		8			7			25		4			30		20	22		33	- 3	36		-	2
19	22		14					70	97	7	7							8	1	18	0,1				5	24	30	25			1	*	33	20	3
31	36	15			35	6			•	10			8	12	2	100	-	9	24	1	22	20			23	33		20			20	0	18	-	Ē
8		24	12			5	3	22	23	18	20	0	4	36	15	31			33	0	34	2	10		16	1	35	19			2	10	26	32	E
		28	9				10	29	34	12	33	23		1			30	25	35			7				4			18	24			8	22	1
20			34		10	31	26	30	9	35	56		29		22		24	3	19		28	32		10	13	36			43		(4)				É
4	11				7	25	17		ĢĒ.		9	28	26		33	150	2.4		30	27		5			6	32		12	8		36			-	3
35	34	33	10		23	21	9	27			32			6	1	8	25	26	12		3			16				10	4	11	15	18	17		1
	29	32	9		36			18	1				13	27		24		20		34	35		0				14	13	21	5		7		8	3
5		8	16	7	ô	12		25	17			25	33	23		(3)	19	27				15	30	11	1	31	32	80	1		26	9.9	13		
17	4	1		1		30	0	36	7	15	2		N.		9	36	28	Š			33	10	8	8	93	V	6	10		25		-7			3
97	15	26		35				18			33	12	10	34		0	36	16		19	C C			0	(K)	22	33	and the last	2			32	-		2
		12		9					35			4	16	11		5	3	6	23					17		15	19	34	25				36	29	3
15	-			35		8		-	26	27	0		21	-						10	7	35	-	-	36				30		17	1		1	
17.7	21	-	11			22	25	12	33		17	10		24	13	32	4	30		29		9	36	20	-	-	-	35		27		15		31	3
25	-	19		17						1		35	-	15	34				32	22	110	14	20		28	-	24				30	21	7	1	8
38	3		28	8		19						20	23	17	17			5		15	18	.6	12			29		27	33	32			14	25	2
32		27	4				-	35	_		24	7		22		14		19	3		1.	ô	25	-	15	-	16	17	35		-	23.	55	9	1
30	12	35	LV.			14	10	32	18	110	1		120	33	U.C.	2	-8	1	22	17	16	23	27	0	25	21			5		13	28			1

• En cada uno de esos puzzles, verificar que la solución es correcta es muy fácil, mientras que encontrar una solución a veces es extremadamente difícil.

En función de "Problemas de decisión"

La clase P como un problema de decisión natural

• Desde este punto de vista, la clase P está compuesta por todos los problemas de decisión que pueden ser resueltos de manera eficiente.

- Un problema de decisión S <u>c</u> {0, 1}* puede ser resuelto eficientemente si:
 - Existe un algoritmo A computable en tiempo polinomial, dado que:
 - Para cada x, A(x) = 1 si y solo si $x \in S$
- Denotaremos como P a este tipo de problemas

En función de "Problemas de decisión"

La clase NP como un problema de decisión natural

La definición más popular indica que:

 Un problema de decisión está en NP si una Máquina de Turing nodeterminista entrega una solución que defina la membresía de una instancia en un grupo.

 Pero podemos ver esta definición de una manera menos ficticia, sin cambiar el significado de la siguiente manera...

La Clase NP y los sistemas de prueba NP

- Decimos que, la membresía de una instancia en un set puede ser determinada a través de una prueba adecuada.
- Entonces, definimos NP como la clase de problemas de cisión que tiene sistemas de pruebas eficientemente verificables.

• En términos generales, decimos que un set **S** tiene un sistema de pruebas si las instancias en **S** tienen pruebas válidas de membresía, mientras que las instancias que no están en **S** no tienen pruebas válidas.

P vs NP

- Si pudiéramos demostrar que todos los problemas en NP pueden ser resueltos de manera eficiente, esto es:
 - NP <u>c</u> P
- Y por lo tanto la **correctitud** de su solución es verificable en tiempo polinómico para cada problema.
- Entonces podemos decir que P = NP

P = NP

- Actualmente no ha sido posible verificar esta igualdad.
- Actualmente se está suponiendo que P ≠ NP
- En sesiones posteriores veremos de qué manera se podría comprobar esta igualdad, y cual es su contexto histórico.

CHECK - OBJETIVOS DE LA SESIÓN

- Conocer los conceptos de clases de problema P y NP
- Relacionar estas clases en cuanto a sus características
- Analizar ejemplos de problemas clasificados en estas clases

CHECK





818181181818

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu Jessica Meza-Jaque – je.meza@uandresbello.edu