



Introducción

Espacio de estados

Tanto en los problemas de decisión de existencia, como en los de optimización, lo que buscamos es un objeto que cumpla con cierta propiedad. Es muy importante que en un problema quede claro cuál es todo el universo válido en donde podemos buscar una solución. No es lo mismo buscar una solución a $x^2 = -1$ en los números reales, que en los números complejos.

Dado un problema algorítmico de los tipos anteriores, al conjunto de donde puede salir la solución le llamamos el **espacio de estados** del problema. Se pueden modelar muchos problemas utilizando estructuras matemáticas conocidas.

Entre las estructuras numéricas que se pueden usar como espacios de estados están:

- Los números naturales.
- Los números enteros.
- Los números racionales.
- Los números reales.
- Los números complejos.
- Los elementos de \mathbb{Z}_n para algún entero n .
- Los elementos de un grupo.
- Para un entero n , los números del 1 al n .

También hay estructuras combinatorias que se pueden usar como espacios de estados:

- Para un entero n , las permutaciones de n elementos.
- Para enteros n y k , todas las posibles elecciones de k elementos de un conjunto.
- Para un entero n , todos los posibles subconjuntos de un conjunto.
- Para enteros n y k , todos los subconjuntos (o combinaciones) de tamaño k de n elementos.

Hay algunas otras estructuras que son geométricas:

- Las configuraciones de n puntos en el plano.
- Las configuraciones de n segmentos en el plano.
- El conjunto de triángulos del plano.
- El conjunto de 2^n vectores en \mathbb{R}^n .

Y como últimos ejemplos también tenemos estructuras de teoría de gráficas

- Todas las posibles gráficas en n elementos.
- Todas las trayectorias de alguna gráfica.
- Todas las formas de dirigir las aristas de una gráfica.

Un problema puede tener más de un espacio de estados que se pueda plantear para resolverlo. Usualmente lo que intentamos hacer es que el espacio de estados sea lo más pequeño o simple posible, pero sin que se pierda la información que necesitamos para resolver el problema.

Veamos algunos ejemplos de problemas concretos y posibles espacios de estados.

Ejemplos de espacios de estados

Problema. Tenemos una lista de (n) números distintos. ¿Qué espacio de estados podemos estudiar para determinar si hay tres de ellos distintos cuya suma sea (2021) ? ¿Qué tamaño tiene asintóticamente? ¿Qué tipo de problema es?

Solución. Este es un problema de decisión, pues nos está preguntando por la existencia de algo.

Pensemos que los números que nos dan son (a_1, a_2, \dots, a_n) . Una forma de plantear el espacio de estados es considerar todas las posibles ternas de elementos:

$$E = \{(a_i, a_j, a_k) : i, j, k \in \{1, \dots, n\}\}$$

Para cada elemento en (E) , podemos verificar que (i, j, k) sean distintos y ver si $(a_i + a_j + a_k = 2021)$. Aquí (i, j, k) pueden tener cada uno (n) posibilidades, así que (E) tiene (n^3) elementos que son $(\Theta(n^3))$.

Otra forma de plantear el espacio de estados es pedir que de entrada los índices (i, j, k) sean distintos, es decir,

$$E' = \{(a_i, a_j, a_k) : i, j, k \in \{1, \dots, n\}, i \neq j, j \neq k, k \neq i\}$$

Si lo hacemos de esta manera, (i) tiene (n) posibilidades, luego (j) sólo $(n-1)$ y (k) sólo $(n-2)$. Así, (E') tiene $(n(n-1)(n-2)) < (n^3)$, pero que aún es $(\Theta(n^3))$. Revisar todas las ternas de (E') es suficiente pues de entrada estamos descartando las que tienen elementos iguales, que no nos interesan. Es un poco mejor pensar el problema con este espacio de estados, pues es más pequeño: aunque tiene el mismo tamaño asintótico, nos puede ahorrar tiempo de ejecución.

Finalmente, una tercer forma de plantear el espacio de estados es considerando únicamente las ternas en donde los índices aparecen en orden:

$$E'' = \{(a_i, a_j, a_k) : i, j, k \in \{1, \dots, n\}, i < j < k\}$$

Aquí el tamaño de (E'') baja a $(\binom{n}{3})$, pues hay un elemento por cada subconjunto de (3) elementos de $(\{1, \dots, n\})$. Aunque reducimos mucho el espacio de estados, de cualquier forma sigue siendo un espacio de estados que cubre todos los casos que nos interesan. Por ejemplo, $(a_1 + a_3 + a_2 = 2021)$ si y sólo si $(a_1 + a_2 + a_3 = 2021)$ por la conmutatividad de la suma. Una propiedad matemática nos ayudó a revisar menos posibilidades.

Tenemos que $(\binom{n}{3} = \frac{n(n-1)(n-2)}{6})$. Si bien sigue siendo $(\Theta(n^3))$, ahora estamos dividiendo entre (6) , lo cual es una buena ventaja comparado con (E') .

Problema. Tenemos una lista de (n) números. ¿Qué espacio de estados hay que estudiar para ver cuáles de ellos tienen (\square) suma más pequeña? ¿Qué tamaño tiene asintóticamente? ¿Qué tipo de problema es?

Solución. Este es un problema de optimización, pues queremos minimizar la suma de algunos números.

No nos están diciendo cuántos números podemos sumar, así que tenemos que cubrir todas las posibilidades. Podemos sumar cero, uno, dos, ..., (n) números. En otras palabras, debemos pasar por todos los posibles subconjuntos. De esta manera, una primer forma de plantear el espacio de estados es como sigue:

$$E = \{\emptyset, \{a_1\}, \dots, \{a_n\}, \{a_1, a_2\}, \dots, \{a_{n-1}, a_n\}, \dots, \{a_1, a_2, \dots, a_n\}\}$$

Si lo hacemos de esta manera, tenemos (2^n) elementos en (E) , que es de orden $(\Theta(2^n))$.

Si nos ocurre ningún otra idea para resolver el problema, de cualquier forma esto ya nos da una solución: para cada elemento en (E) verificamos la suma de sus elementos. En cada momento vamos guardando lo mejor en una variable `minimo`, y si el mínimo se mejora, lo actualizamos. Si además quisiéramos guardar el conjunto que da la mejor suma, lo vamos almacenando en una variable `arg_minimo`, que también se actualiza cuando el mínimo se mejora.

Pasemos al segundo problema. Queremos ver cuáles de los números dados tienen la menor suma posible. No nos están (\square)

diciendo cuántos números podemos tomar, así que, de entrada, el espacio de estados de nuestro problema son todos los posibles subconjuntos de los $\binom{n}{k}$ elementos que tenemos, es decir:

$$E = \{\emptyset, \{a_1\}, \{a_2\}, \dots, \{a_1, a_2\}, \dots, \{a_1, a_2, \dots, a_n\}\}$$

Este espacio de estados tiene tamaño 2^n , que en notación asintótica justo es de tamaño $\Theta(2^n)$.

(\square)

Espacios de estados más complicados

En los ejemplos anteriores, los espacios de estados fueron muy «conocidos» o «simples» en el sentido de que son objetos matemáticos que se estudian muy seguido. Pero esto no necesariamente es así en todos los problemas. Hay problemas con espacios de estados más complicados o más particulares.

Ejemplos.

- Para resolver el problema del Sudoku, el espacio de estados se parece a rellenar entradas de una matriz, pero hay restricciones dadas por las reglas del Sudoku que son difíciles de poner en términos matemáticos muy estándar.
- Para resolver el problema de si se pueden colocar 3 torres de ajedrez, 3 caballos y 4 alfiles sin que se ataquen, también debemos de considerar un espacio de estados muy raro dado por las reglas de ataque de las piezas.
- Para resolver problemas de empaquetamiento de objetos, es posible que podamos plantear una gran parte del espacio de estados en términos matemáticos, sin embargo, hay muchas restricciones de la vida real que en la práctica se tienen que incluir, por ejemplo:
 - Que los objetos cerca de los bordes de la caja no deban ser filosos para no romper la caja.
 - Que los objetos no queden cerca de lugares donde se calienta mucho.
 - Que los objetos frágiles no queden cerca de objetos duros.
 - Que no se exceda cierto peso dentro de la caja.

(\square)

Heurísticas para creación de algoritmos

A grandes rasgos, una **heurística** es una herramienta que nos ayuda pensar cómo resolver un problema. Muchas veces comenzamos con un problema del cual tenemos poco contexto y queremos saber cómo comenzar a atacarlo. Las heurísticas guían esta resolución.

En el caso de los problemas algorítmicos, las heurísticas nos pueden guiar en cómo crear algoritmos sin tener que empezar desde cero. Las heurísticas que veremos en las siguientes entradas son:

- Exploración exhaustiva
- Recortes al espacio de estados
- Algoritmos voraces
- Divide y conquista
- Recursión y teorema maestro
- Backtrack en búsquedas combinatorias
- Más ejemplos de backtrack
- Programación dinámica
- Métodos probabilistas

Tarea moral

Los siguientes problemas te ayudarán a practicar lo visto en esta entrada. Para resolverlos, necesitarás usar herramientas matemáticas, computacionales o ambas.

computacionales y amos.

1. Problema
2. Problema
3. Problema
4. Problema
5. Problema



[Anterior](#)

[Tipos de problemas algorítmicos](#)

[Siguiete](#)

[Exploración exhaustiva](#)



Por Leonardo Ignacio Martínez Sandoval

© Copyright 2022.