Exploración exhaustiva

Introducción

La exploración exhaustiva es la primer heurística que estudiaremos. Consiste en resolver un problema mediante la exploración total de un espacio de estados para encontrar un valor óptimo o un objeto con una propiedad deseada.

Es una técnica básica, pero en ocasiones es la única con la cual contamos. Típicamente es la única opción en problemas en donde tenemos muy poca estructura matemática que podamos explotar. También hay problemas en donde queremos garantizar pasar por todas las posibilidades.

A la exploración exhaustiva también se le conoce como «fuerza bruta».

Problemas a resolver

Usaremos la heurística para resolver los siguientes problemas. No necesariamente será la mejor forma de resolverlos, pero nos ayudará a entender el concepto que estamos estudiando. Antes de comenzar a leer cómo resolvemos los problemas, te sugerimos que los leas para entender en qué consistes y que intentes resolverlos por tu cuenta.

Problema ¿De cuántas formas se puede poner a $10,000$ como suma de cuadrados de dos números enteros positivos? ¿En cuál de las expresiones $x^2+y^2=10000$ se minimiza $3x+5y-1$?

Problema ¿Cuáles son las palabras más largas en español que tengan únicamente cuatro vocales? La lista de palabras que nos interesa estudiar está en el archivo `espanol.txt` .

Problema Las letras a,b,c,d,e,f,g,h,i,j representan dígitos distintos. ¿para cuántas elecciones tenemos que $\frac{abcde}{fghij}$ es un número entero? Aquí $abcde$ y $fghij$ son los números de cinco dígitos obtenidos de concatenar los dígitos correspondientes.

Sumas de cuadrados y minimizar expresión

Comencemos con la solución del siguiente problema.

Problema ¿De cuántas formas se puede poner a $10,000$ como suma de cuadrados de dos números enteros positivos? ¿En cuál de las expresiones $x^2+y^2=10000$ se minimiza $3x+5y-1$?

Antes de comenzar una exploración exhaustiva, es importante determinar cuáles son todas las posibilidades que debemos cubrir. Una forma de resolver el problema es plantear como espacio de estados un cierto conjunto de parejas (x,y) y preguntarnos cuándo $x^2+y^2=10000$. Observa que si $x\geq 100$ ó $y\geq 100$, entonces la expresión izquierda excede a la derecha, así que basta limitarse a $1\leq x \leq 100$ y $1\leq y \leq 100$.

Una manera de realizar la exploración de todas las parejas cuyos valores de x y y satisfacen dichas desigualdades es mediante el uso de dos ciclos anidados. La implementación quedaría de la siguiente manera.

```
# Primero, haremos una exploración exhaustiva para ver
# quienes son todas las parejas.
cuantos=0
cuales=[]
for x in range(1,100):
    for y in range(1,100):
        if x**2+y**2==10000:
            cuantos+=1
            cuales.append((x,y))

print("Hay {} parejas. La lista de parejas es {}.".format(cuantos, cuales))
```

```
Hay 4 parejas. La lista de parejas es [(28, 96), (60, 80), (80, 60), (96, 28)].
```

Ya que tenemos todas las parejas que cumplen, ahora podemso hacer una exploración exhaustiva ya únicamente dentro de esas soluciones para determinar en cuál se minimiza la expresión \(3x+5y-1\).

```
# Ahora, hagamos otra exploración para ver en cuál se minimiza 3x+5y-1
minimo=10000000 # Fijamos un mínimo grande

for pareja in cuales:
    x=pareja[0]
    y=pareja[1]
    if 3*x+5*y-1<minimo:
        minimo=3*x+5*y-1
        optimo=(x,y)

print("El mínimo es {} y se alcanza con la pareja {}.".format(minimo,optimo))
```

```
El mínimo es 427 y se alcanza con la pareja (96, 28).
```

Podríamos estudiar este problema de manera más general, buscando dos números \(x\) y \(y\) cuya suma de cuadrados sea \(n\). En este caso, la estrategia anterior nos daría que \(x\) y \(y\) están en \(\{1,2,\ldots,\lfloor \sqrt{n} \rfloor-1\}\).

Así, en el ciclo externo corremos por \(\Theta(\sqrt{n})\) posibilidades y en el interno también. De este modo, estos ciclos anidados corren en total en tiempo \(\Theta(n)\). Siendo un poco más cuidadosos, podemos hacer que tan sólo con esta parte del código también podamos minimizar la expresión \(3x+5y-1\).

Palabras largas con pocas vocales

Resolvamos ahora el siguiente problema.

Problema ¿Cuáles son las palabras más largas en español que tengan únicamente cuatro vocales? La lista de palabras que nos interesa estudiar está en el archivo `espanol.txt` .

Lo primero que haremos es una función auxiliar que cuenta vocales. También funcionará por exploración exhaustiva, pues para cada palabra, revisaremos cada una de sus letras, en búsqueda de vocales.

```
# Función auxiliar para contar vocales.
def contarvocales(palabra):
    # Lista de vocales con posibles símbolos especiales.
    vocales='aeiouáéíàèìòùóúüAEIOUÁÉÍÓÚÜÀÈÌÒÙ'
    cuenta=0
    for j in palabra:
        if j in vocales:
            cuenta+=1
    return cuenta

# Algunas pruebas para ver que contarvocales hace lo que queremos
print(contarvocales('Hola mundo!'))
print(contarvocales('Esta oración tiene acentos'))
print(contarvocales('PIngÜinoS y MurCIElAGOs'))
```

```
4
12
9
```

Ahora, para contar cuántas palabras cuentan con la propiedad deseada, lo que haremos es usar esta función auxiliar en cada una de las palabras de la lista que nos dan. Esto está haciendo una búsqueda exhaustiva pues pasamos por todas ellas.

```
lista=open('espanol.txt','r',encoding = "ISO-8859-1")
linea=lista.readline()
maximo=0

# Vamos a encontrar todas las palabras que cumplan tener cuatro vocales.
# Cada que encontremos una, compararemos su longitud con la mejor longitud
# encontrada hasta ahora. Si es mejor, tenemos que reiniciar nuestra cuenta
# de cuáles son
```

```
# de cuáles son.
mejores=[]
while linea:
    # Algunas líneas del archivo 'subj' al final. Para limpialas hacemos
    # lo siguiente.
    limpio=linea[:-1].split(' ')[0]
    # Ahora sí, procesamos la cadena limpia. Típico algoritmo que va
    # almacenando los y se reinicia si encuentra algo mejor.
    if contarvocales(limpio)==4:
        if len(limpio)>maximo:
            maximo=len(limpio)
            mejor=limpio
            mejores=[mejor] # Reinicia la lista y pone a la nueva palabra.
        elif len(limpio)==maximo:
            mejores.append(limpio)
    linea=lista.readline()

print('''Las palabras con cuatro vocales y la máxima cantidad de letras tienen {} letras. \nSon las siguientes {}:'''.format(maxi
for palabra in mejores:
    print('- '+palabra)

lista.close()
```

```
Las palabras con cuatro vocales y la máxima cantidad de letras tienen 14 letras.
Son las siguientes 4:
- prescriptibles
- transformables
- transpondremos
- transportables
```

Fracciones enteras

Finalmente, veamos cómo resolver el siguiente problema.

Problema Las letras \(a,b,c,d,e,f,g,h,i,j\) representan dígitos distintos. ¿para cuántas elecciones tenemos que \(\frac{abcde}{fghij}\) es un número entero? Aquí \(abcde\) y \(fghij\) son los números de cinco dígitos obtenidos de concatenar los dígitos correspondientes.

Tarea moral

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

1. Se toman tres enteros \(a\), \(b\) y \(c\) en el intervalo \([1,100]\). ¿Para cuáles de ellos el valor de \(a^2+2b^2+3c^2-2ab-5bc-7ca\) es mínimo? Si \(a,b,c\) están ahora en el intervalo \([1,n]\), ¿en cuánto tiempo corre tu algoritmo?
2. ¿Existe alguna palabra en español que use exactamente diez vocales y diez consonantes? Si sí, ¿cuántas hay? La lista de palabras que nos interesa estudiar está en el archivo `espanol.txt`. Plantea un problema general y estudia la complejidad de tu algoritmo.
3. Encuentra todas las palabras en español que usen cada vocal exactamente una vez. De entre ellas, ¿cuál es la más larga? ¿Cuál es la más corta? Generaliza el problema y estudia la complejidad de tu algoritmo.
4. Problema
5. Problema

Anterior
Espacios de estados y heurísticas

Siguiente
Recortes al espacio de estados

Por Leonardo Ignacio Martínez Sandoval