Programación Dinámica: Mochila

Guillermo I. Bautista G. *Estudiante, UPIIZ*

Resumen—La solución de problemas se puede clasificar según el problema, y se pueden encontrar problemas de soluciones rápidas, llamados en programación problemas logarítmicos que a pesar de la cantidad de entradas no tardan mas que $log(C)$ en resolverse, y por otro lado tenemos problemas que son tan difíciles de clasificar que los tenemos apartados en una sección denominan NP (no polinomiales) esto quiere decir que son problemas de demora exponencial o algo más tardado. se atacará un problema famoso de este tipo según distintas formas de programación llegando a una solución propuesta.

Index Terms—Mochila, algoritmo, dinámico, item.

1. Introduction

La programación dinámica puede ser una herramienta útil incluso en casos donde no se sepa exactamente como se debe resolver un problema, dado a que uno de sus principios es la modularidad, se encarga de subdividir un problema en problemas más pequeños a resolver que en un futuro serán de necesarios para la solución final.

2. Mochila de beneficios

Dada una mochila de capacidad k y una lista de items con n beneficios, qué combinación de items me da el mayor beneficio.

3. Generador de lista de items

Según la descripción del problema tenemos que considerar que en una solucion habra la capacidad de una mochila, una lista de items que sera el mercado y otra lista de items que serán los que deban ir en la mochila, estos items tambien generaran un beneficio maximo y por el hecho de pensar en soluciones anteriores ocuparemos una matriz de beneficios. dado lo que necesitamos, lo más complejo de conseguir es esa lista de items totales, por suerte, conocemos que un item se compone de un peso y un beneficio, por lo tanto podemos generarlo de esa manera

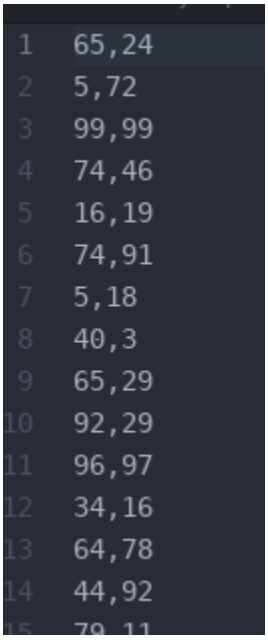

Figura 1. Lista generada de forma aleatoria

Nosotros así como creamos nuetra lista, podemos leerla para que sea utilizable por el problema, y así es como se realiza.

4. Resultados

Dado a que este código fue en su mayor parte visto en clase lo único que se

- *Unidad Interdisciplinaria de Ingenierías campus Zacatecas*
- *Instituto Politécnico Nacional*

```
run:
[65,24, 5,72, 99,99, 74,46, 16,19, 74,91
3,70
1,6
5,72

BUILD SUCCESSFUL (total time: 1 second)
```

Figura 2. Utilizando una mochila de peso 10, este es nuestro mayor beneficio

puede sacar de enseñanza aparte es cómo manipular la mochila, con el simple hecho de que podemos darle un peso de 100 para ver como cambian los artículos, codigo github: https://github.com/Memotets/AdA/tree/master/src/Mochila:

```
run:
[65,24, 5,72, 99,99, 74,46, 16,19, 74,91, 5,
14,74
3,70
5,67
9,46
1,6
11,82
15,98
9,65
3,16
24,83
5,72

BUILD SUCCESSFUL (total time: 0 seconds)
```

Figura 3. Utilizando una mochila de peso 100

5. CONCLUSIONES

A diferencia del TSP aquí se muestra una solución puramente dinámica, lo que implica que guardamos toda la información necesaria para llegar a la siguiente solución y a su vez esta sirve para dar una nueva solución, y por el hecho de que esta planeado para que sea de esa forma, debemos de considerar las estructuras utilizadas ya que todo tiene un proposito aunque no parezca obvio o sea complicado de visualizar como la matriz de beneficios que es la que nos hace distinguir entre que es mejor para una mochila de 10 y que lo es para una de 100.