

UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO

FACULTAD DE INGENIERIA Y CIENCIAS DE LA COMPUTACION Y TELECOMUNICACIONES

INTELIGENCIA ARTIFICIAL

CONTENIDO: EL PROBLEMA DE LA MOCHILA

PORCENTAJE TERMINADO: 50%

INTEGRANTES	DT	HG	HI	EVAL
Flores Veizaga Eudenia Gandira		1	1	70
Garcia Taboada Brayan Albaro		1	1	70

Fecha de presentación : Jueves, 14 de Marzo de 2024

Fecha Presentada : Jueves, 14 de Marzo de 2024

Días de Atraso : 0

TRABAJO EN PAREJAS.

Se tiene n -objetos con pesos p_1, p_2, \dots, p_N . Se tiene también una mochila que puede transportar como máximo MAX , de peso. Encontrar todas la combinaciones de objetos posibles que se pueden transportar en la mochila.

Ingresar los pesos de objetos en una Lista de Enteros L_1 y mostrando la combinación de objetos en una Lista de Enteros L_2 y utilizando un algoritmo de la forma de llamada recursiva dentro de un ciclo.

[]

[2]

[2, 4]

[2, 5]

[4]

[4, 5]

[5]

Para cada uno de los siguientes problemas, además mostrar la cantidad de soluciones posibles.

1. Encontrar todas las combinaciones de pesos de objetos que se pueden transportar en la mochila y la cantidad de soluciones posibles.

[]

[2]

[2, 2]

[2, 2, 5]

[2, 5]

[2]

[2, 5]

[5]

Combinaciones: 8

2. Encontrar todas las combinaciones de pesos diferentes que se pueden transportar en la mochila. (Se asume que existen objetos con pesos iguales)

3. Encontrar todas las combinaciones de pesos entre los pesos a y b inclusive que se pueden transportar en la mochila.

4. Encontrar las combinaciones de m-objetos ($m \leq n$), que se pueden transportar en la mochila.

[]

[1]

[1, 2]

[1, 2]

[1]

[1, 2]

[1]

[1]

ANEXOS

Ejercicio

```
public class Pruebas {  
    public static void main(String[] arg) {  
        //LinkedList<Integer> L1 = new LinkedList<>();  
        //sumandos (L1, 4, 1);  
        LinkedList<Integer> L1=new LinkedList();  
        LinkedList<Integer> L2=new LinkedList();  
        L1.add(2);  
        L1.add(4);  
    }  
}
```

```

        L1.add(5);
        mochila(L1,L2,10,0);
    }
    private static void mochila(LinkedList<Integer> L1,
LinkedList<Integer> L2, int max, int i) {
        int sum = suma(L2);
        if(sum>max)
            return;
        System.out.println(L2);
        int j=i;
        while(j<L1.size()){
            L2.add(L1.get(j));
            mochila(L1,L2,max,j+1);
            L2.removeLast();
            j=j+1;
        }
    }

    private static int suma(LinkedList<Integer> L2) {
        int sum=0;
        for (Integer integer : L2) {
            sum+=integer;
        }
        return sum;
    }
}

```

1.

```

public static void main(String[] arg){
    //LinkedList<Integer> L1 = new LinkedList<>();
    //sumandos(L1,4,1);
    LinkedList<Integer> L1=new LinkedList();
    LinkedList<Integer> L2=new LinkedList();
    L1.add(2);
    L1.add(2);
    L1.add(5);
    int cant= mochila(L1,L2,10,0);
    System.out.println("Combinaciones: "+cant);
}
private static Integer mochila(LinkedList<Integer> L1,
LinkedList<Integer> L2, int max, int i) {
    int sum = suma(L2);
    int cant = 0;
    if(sum>max) return cant;
    System.out.println(L2);
    cant++;
    int j=i;
    while(j<L1.size()){
        L2.add(L1.get(j));
        cant+= mochila(L1,L2,max,j+1);
        L2.removeLast();
        j=j+1;
    }
    return cant;
}

private static int suma(LinkedList<Integer> L2) {
    int sum=0;
    for (Integer integer : L2) {

```

```

        sum+=integer;
    }
    return sum;
}

```

4.

```

public class Pruebas {
    public static void main(String[] arg){
        LinkedList<Integer> L1=new LinkedList();
        mochila(L1,1,5,10,0);
    }
    private static void mochila(LinkedList<Integer> L1,int a,int b,
int max, int i) {
        int sum = suma(L1);
        if(sum>max)
            return;
        System.out.println(L1);
        int j=i;
        while(j<(b-a)){
            L1.add(a);
            mochila(L1,a+1,b,max,j+1);
            L1.removeLast();
            j=j+1;
        }
    }

    private static int suma(LinkedList<Integer> L2) {
        int sum=0;
        for (Integer integer : L2) {
            sum+=integer;
        }
        return sum;
    }
}

```