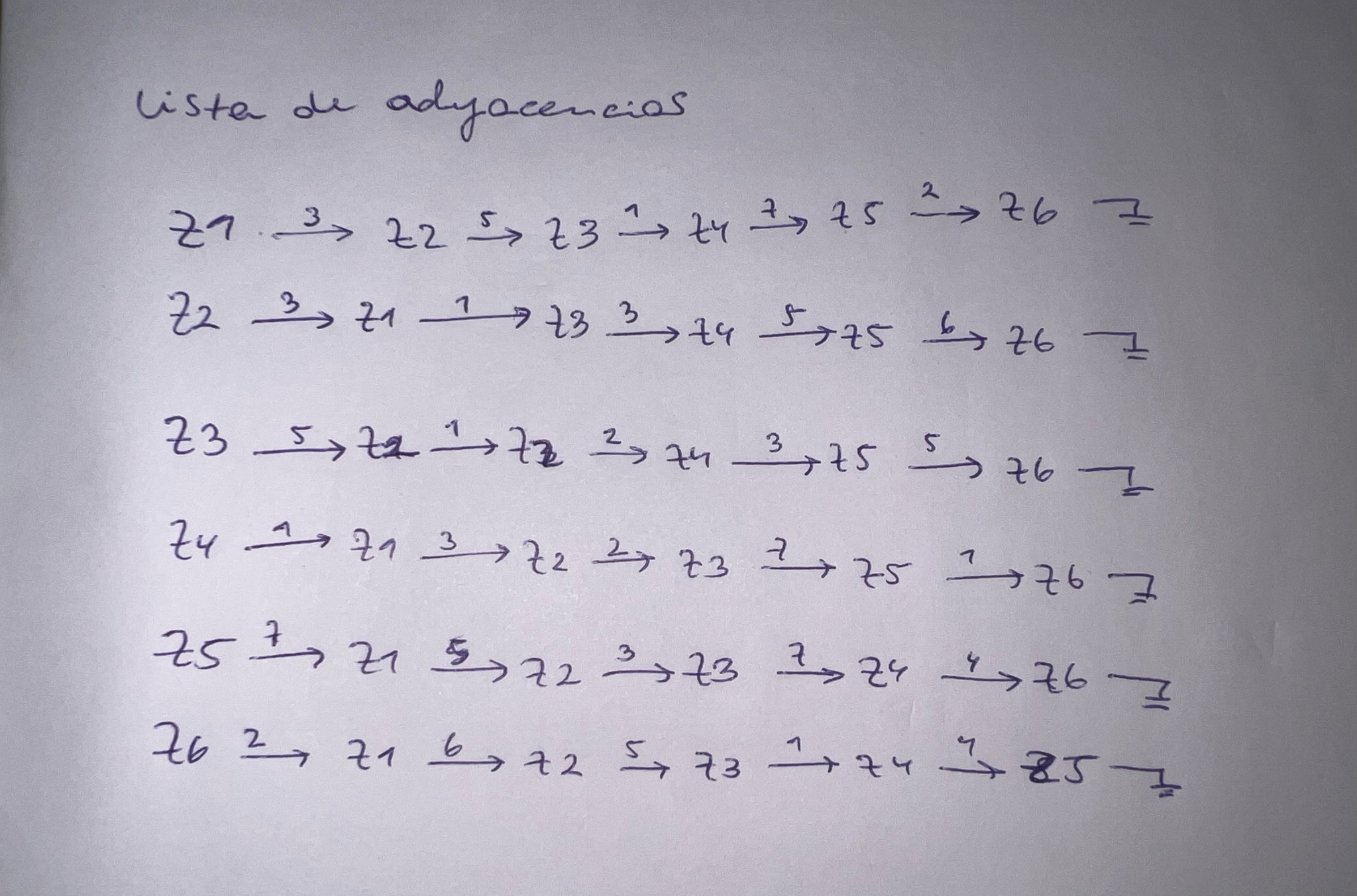
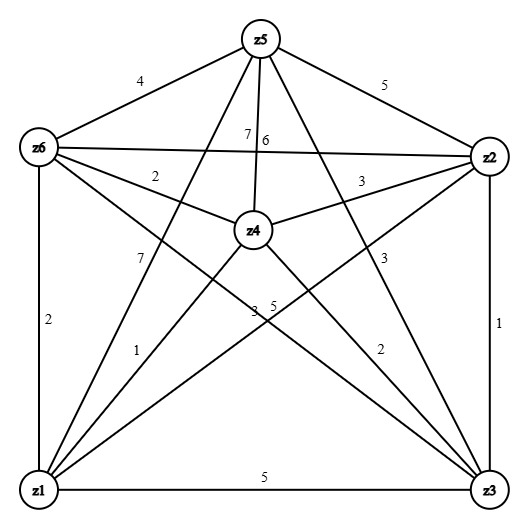
UT8 TA1

### **EJERCICIO 1**

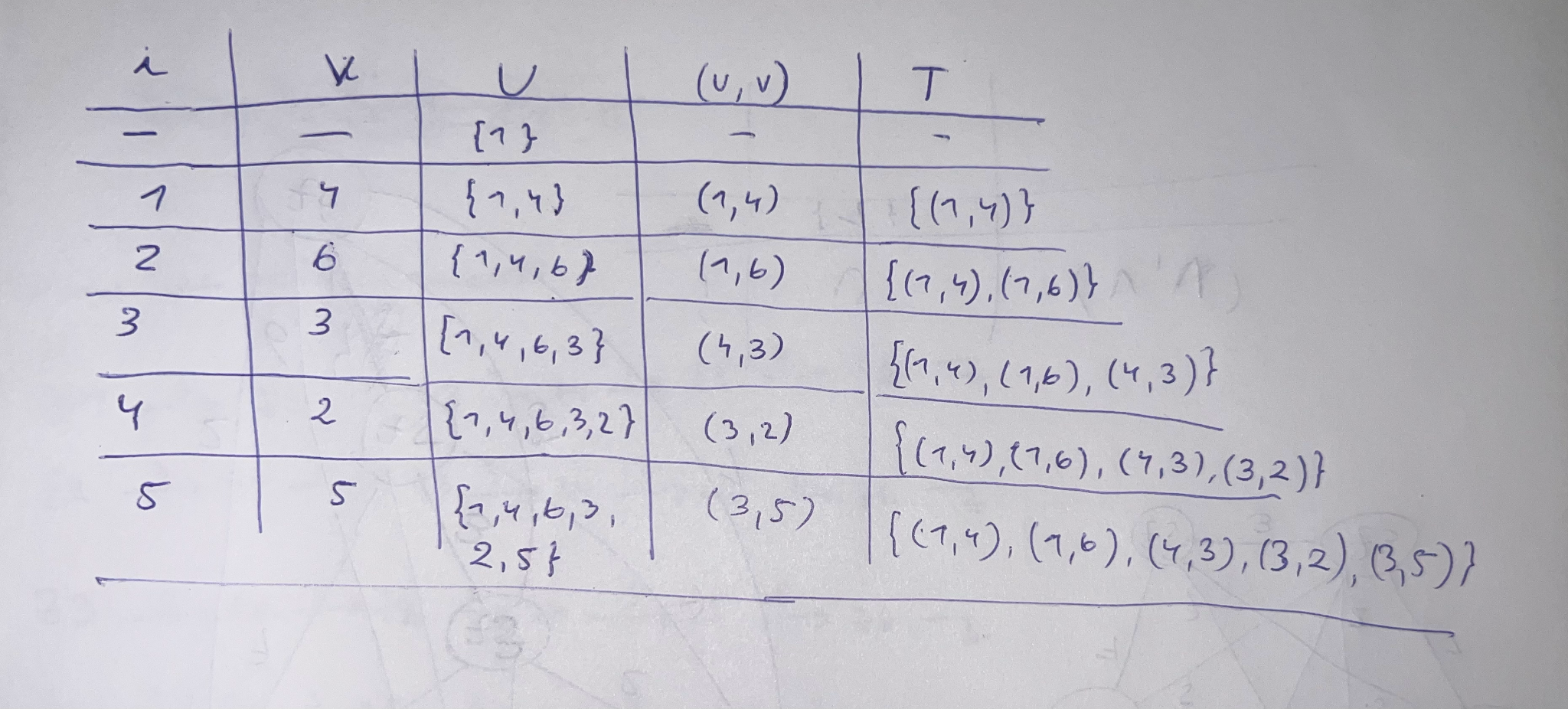
Representa la estructura elegida mediante Lista de adyacencias



Representa gráficamente la red de zonas y distancias.



Determina la mejor opción para la construcción de los caminos de acuerdo con el escenario



### **EJERCICIO 2**

#### **Parte1**

Método TGRAFO.Prim (conjuntoAristas T){

U: vértices [ ];

u, v vértices;

contador = 0

minCosto = maxInt

TGrafo representado por una colección de vértices V y de aristas A donde V hereda de HashSet de tipo TVertice/Comparable

u = V.values.first

U.Agregar(u);

Mientras (contador < ) hacer:

para cada u en U

para cada v en V

si v no esta u

tempArista = Aristas.buscar(u,v)

si (tempArista <> null){

si (tempArista.costo < minCosto) {

minArista = tempArista;

minCosto = tempArista.costo

T.Agregar(minArista);

U.agregar(v)

}

}

}

}

}

}