

## Tarea 3 - Análisis de Algoritmos y Estructura de Datos

Leonardo Iván Espinoza Ortiz  
Universidad de Santiago de Chile  
2-2022

### 1- Introducción

Dos estudiantes aficionados a los juegos de tipo Escape Room están testeando un juego que será comercializado próximamente.

El juego tiene distintos niveles. En cada nivel parte con cierta cantidad de unidades de energía que se van utilizando a medida que se avanza en el juego.

El objetivo del juego es entrar por la puerta inicial y avanzar por diferentes sectores hasta llegar a la salida.

Entre sectores es posible que se encuentren paredes que impiden el paso, y en cada sector se va perdiendo unidades de energía, ya sea por tener que resolver acertijos, enfrentarse con algún obstáculo que debe eliminar, etc.

Cada nivel del juego debe ser completado uno a uno. Lo que deben realizar como testadores es finalizar cada nivel con la mayor cantidad de unidades de energía disponibles para enfrentar el siguiente nivel de la mejor manera. Además, como incentivo se les entregará un bono por cada unidad de energía que tengan al final del juego.

Como estrategia crearon un mapa que representa cada nivel del juego, con la información de los lugares y la cantidad de unidades de energía que se requiere en cada sector de éste, dependiendo de los obstáculos que debe enfrentar.

Cada número en el mapa indica la cantidad de unidades de energía que al menos se debe utilizar en el sector.

Cada mapa tiene únicamente dos puertas: una puerta inicial y una puerta final, que es la entrada al siguiente nivel con un mapa diferente. Cada mapa (uno por nivel) es de la

siguiente

manera:

Puerta Inicio					
5	1	10	1	1	6
8	1	1	1	1	1
2	1	3	4	2	5
8	2	1	1	1	1
3	1	1	1	3	16
1	1	1	1	4	1
Puerta Fin					

Figura 1: Representación del mapa

Puerta Inicio					
5	1 ● ●	10	1 ● ●	1 ●	6
8	1 ● ●	1 ● ●	1 ● ●	1 ●	1
2	1 ●	3 ●	4 ●	2 ●	5
8	2 ●	1 ●	1 ●	1 ●	1
3	1 ●	1 ●	1 ● ●	3 ●	16
1	1 ● ●	1 ● ●	1 ● ●	4	1
Puerta Fin					

Figura 2: Ejemplo de recorrido en el mapa.

Tras esto, se recurre a nosotros ya que conocemos de nuestro conocimiento experto en solucionar problemas rápidamente usando algoritmos y estructuras.

El objetivo de esta tarea es encontrar la mejor estrategia de juego en cada nivel, de tal manera de contar con la mayor cantidad de energía al final de cada mapa, para que pueda ser usada en el siguiente mapa, modelando como grafo.

Para la resolución de esta tarea, se utilizará lo aprendido en la unidad 3 de la asignatura Análisis de Algoritmos y Estructuras de Datos y se creará dicha solución en el lenguaje de programación C.

### 2-Solución propuesta

El enfoque para solucionar este problema se basará en estructuras llamadas grafos, se interpretará la imagen del mapa para modelar un grafo y así introducirlo como archivo de entrada para aplicar el algoritmo para solucionar el problema.

### Tarea 3 - Análisis de Algoritmos y Estructura de Datos

Leonardo Iván Espinoza Ortiz  
Universidad de Santiago de Chile  
2-2022

Se utilizará el mapa de ejemplo para modelar el grafo y las especificaciones para resolver el problema:

-Vértices= 38

-Aristas= 80

-Pesos= Sí

-Dirigido o no Dirigido= Dirigido.

-Representación a utilizar: Matriz de adyacencia

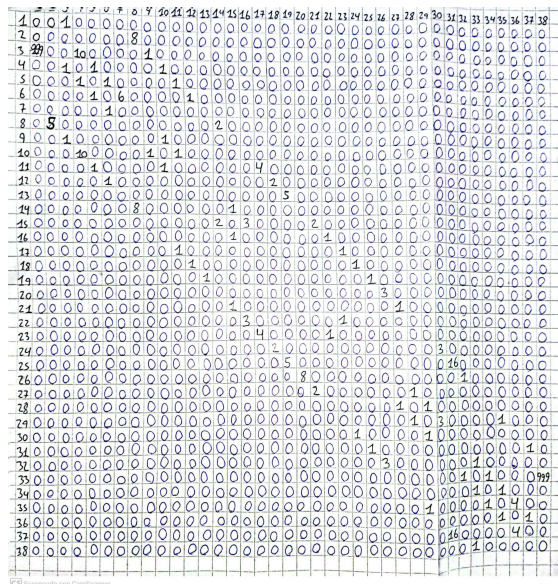


Figura 3: Matriz de adyacencia

-Representación en grafo:

Luego de crear la matriz, se procede a crear el grafo a representar.

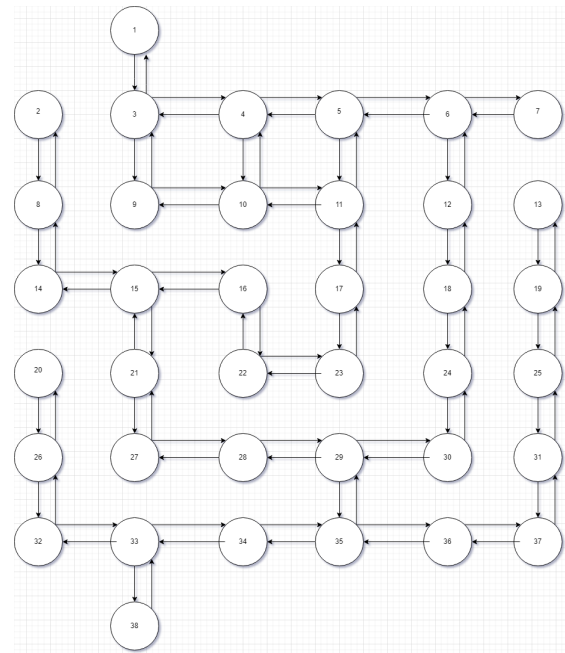


Figura 4: Representación de los mapas a través de un grafo.

-Vértices están numerados de izquierda a derecha para un mayor orden

Se agregan 2 vértices adicionales a la representación del mapa.

El vértice inicial que será la “puerta de entrada” ya que permite calcular la energía del primer nivel.

El vértice final, que únicamente permite identificar la puerta de salida al ser el último vértice creado.

-La solución propuesta será aplicar Camino Mínimo (dijkstra)

-El programa se centra en el desarrollo principal de 2 funciones implementadas:  
Ruta recorrida y energía gastada.

a-Energía gastada:

Función que permite calcular la cantidad de energía gastada dado un mapa ingresado con sus respectivos niveles de gasto de energía.

```
int* Energiagastada(TDAgrafo* grafo, int inicio,int energia)
{
```

Figura 5:función energía gastada

## Tarea 3 - Análisis de Algoritmos y Estructura de Datos

Leonardo Iván Espinoza Ortiz  
Universidad de Santiago de Chile  
2-2022

Esta función recibe el grafo leído, un vértice de inicio que marca el punto de inicio y la cantidad de energía gastada.  
El principal fundamento de esta función es el algoritmo de Dijkstra.  
Se aplica dijkstra en el grafo y luego de esto, Se obtiene la energía total gracias a los últimos valores de los arreglos respectivos de Dijkstra. El antecesor y la distancia.  
Luego de esto, se les resta la energía ingresada al inicio del programa.

b-Ruta Recorrida:

Función que toma el arreglo de antecesores, lo invierte y crea la ruta recorrida asignando los niveles recorridos a través de ciclos para luego ser mostrado en pantalla.

### 3-Resultados y análisis

Tras implementar el código, se logró el resultado esperado.

```
la energia ingresada es
100
la energia restante es
83
La ruta recorrida es:
3 9 10 11 5 6 12 18 24 30 29 35 34 33
```

Figura 6: Resultado de ejecución del grafo modelado anteriormente

```
la energia ingresada es
100
la energia restante es
92
La ruta recorrida es:
2 3 5 7 8
```

Figura 7: Resultado de ejecución de grafo ejemplo

```
la energia ingresada es
100
la energia restante es
94
La ruta recorrida es:
2 3 4 5 6 7
```

Figura 8: Resultado de ejecución de grafo ejemplo 2

```
la energia ingresada es
100
la energia restante es
90
La ruta recorrida es:
2 3 7 10
```

Figura 9: Resultado ejecución grafo ejemplo 3  
Se puede observar una buena ejecución del programa realizado.

También se puede ver el siguiente tiempo de ejecución:

### 4-Conclusiones

Tras realizada la tarea, se puede concluir que se aprendió y se cumplió el objetivo principal de este problema.

Se aprendió sobre más formas de implementar grafos( como el grafo direccionado ponderado) y se aprendió de la importancia y buen uso del algoritmo de Dijkstra.

### Anexo manual de usuario

Se debe ingresar por consola el archivo inicial y las unidades de energía.

```
3>
3>tarea GrafoMST.in 100|
```

Figura 7: Ejemplo de Ingreso por consola para la ejecución del programa