Atravesar el laberinto

Objetivo.

- Aprender a implementar el algoritmo A*.
- Aprender a usar el tipo std::priority_queue.
- Aprender a usar el tipo std::tuple.

Descripción.

El objetivo de la práctica es realizar un programa que dado un laberinto (representado por un grafo) encontrar el camino mínimo que une la entrada con la salida si lo hay.

Para ello vamos a utilizar el algoritmo A* que, como sabes, se puede implementar como una modificación del algoritmo de Dijkstra utilizando una función heurística que nos de una aproximación de la distancia que nos queda para llegar al destino. Nosotros vamos a utilizar la distancia euclídea para obtener esta aproximación.

El algoritmo A* utiliza una cola de prioridad. Se deberá utilizar el tipo std::priority_queue de la STD de C++ para implementarla. Los elementos insertados en la cola será tuplas <distancia-origen+estimación al destino, distancia-origen, vértice-a-añadir, vértice predecesor>. Para implementar estas tuplas se deberá utilizar el tipo std::tuple, que es una generalización del tipo std::pair.

El laberinto es una rejilla de puntos coordenados (x, y) con enlaces indicando para cada punto de la rejilla los posibles caminos al siguiente punto. Para representar el laberinto usaremos un grafo no dirigido, donde cada vértice representa un punto de la rejilla. La etiqueta de un vértice codifica las coordenadas del punto (x, y) correspondiente del laberinto de forma que si las dimensiones del laberinto son (10, 10) y la etiqueta de un vértice v es 14, este vértice está codificando el punto (4, 1) ya que 14/10=1 y 14%10=4.

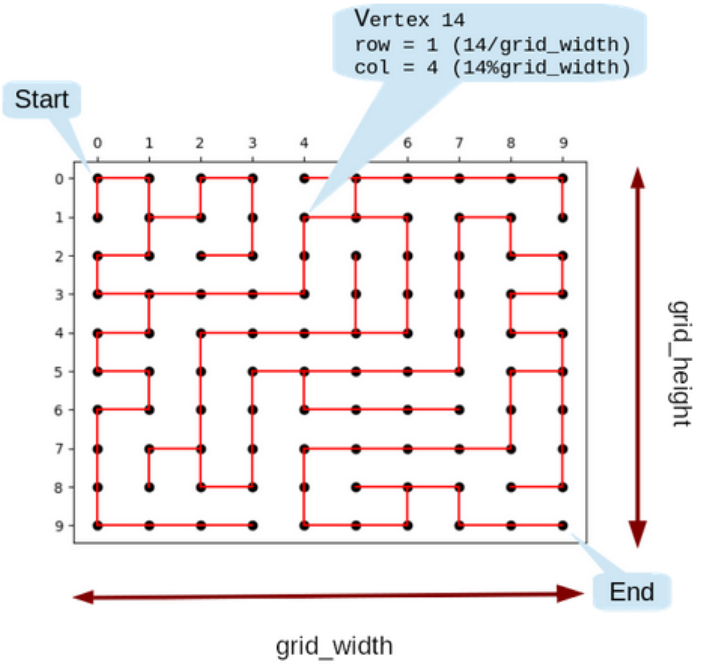

Fig 1. Descripción de un laberinto usando un grafo no dirigido. Un laberinto se define sobre una rejilla de tamaño (alto, ancho) puntos coordenados y los enlaces entre los mismos. En el ejemplo, desde el punto (0,0) podemos ir al punto (0,1) o al punto (1,0). En el grafo no dirigido correspondiente, el vértice 0 estará conectado con los vértices 1 y 10.

Referencias.

https://en.wikipedia.org/wiki/A*_search_algorithm

http://www.cplusplus.com/reference/queue/priority_queue/?kw=priority_queue

http://www.cplusplus.com/reference/tuple/tuple/