**Algoritmo de Dijkstra**

**Idea intuitiva**

El algoritmo de dijkstra determina la ruta más corta desde un nodo origen hacia los demás nodos para ello es requerido como entrada un grafo cuyas aristas posean pesos.

**(Descripción de objetos y funciones)**

**Definición.** Sean G y Gr dos grafos, http://scielo.sld.cu/img/revistas/rii/v38n1/e1504117.jpgun conjunto de reglas de reescritura de grafos, se puede afirmar Gr que  es un grafo reducido a partir de G si al aplicar las reglas de reescritura especificadas en R al grafo G se obtiene el grafo G.

A partir de lo anterior, se enuncia un algoritmo de reducción de grafos sin pérdida de información relevante para la búsqueda de caminos mínimos. Posteriormente, se presenta una modificación al algoritmo de Dijkstra para realizar búsqueda de caminos mínimos en grafos reducidos con el algoritmo de reducción propuesto.

**Reducción de grafos**

La reducción de grafos se realiza utilizando el Algoritmo 1, en este epígrafe se presenta dicho algoritmo y se expone una breve explicación del mismo.

En el caso del algoritmo de reducción que se propone, se debe garantizar que no existan vértices reducidos  que sean adyacentes con el objetivo de poder obtener un camino óptimo de igual costo al obtenido en una búsqueda en el grafo original, para contribuir al logro de este objetivo, se enuncia la siguiente definición:

**Definición**. Sea un grafo G = (V, E) y una relación de equivalencia RE sobre V, un vértice http://scielo.sld.cu/img/revistas/rii/v38n1/e0604117.jpg es interior si http://scielo.sld.cu/img/revistas/rii/v38n1/e1604117.jpg, tal que vi y vj son adyacentes, se cumple que RE(vi , vj)(o sea, ([vi] = [vj]). Un vértice  http://scielo.sld.cu/img/revistas/rii/v38n1/e0604117.jpg es exterior si  o http://scielo.sld.cu/img/revistas/rii/v38n1/e1704117.jpg.

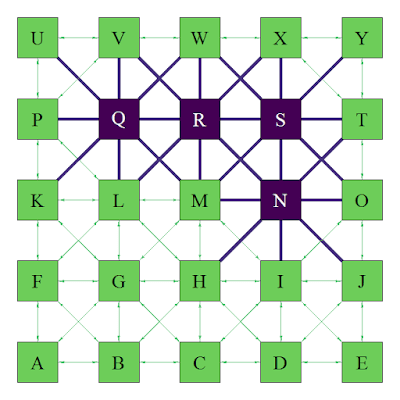
Si la entrada del algoritmo es una relación de equivalencia, lo primero que se hace es obtener una partición a partir de dicha relación. Para  crear  esta partición  se puede seguir una estrategia voraz que divida el conjunto de vértices V  en subconjuntos disjuntos según RE. La estrategia consistiría en comparar cada elemento del conjunto V con los restantes elementos para determinar cuáles están relacionados (según RE) y a partir de estos crear las clases que forman la partición. Si una clase está formada por un solo elemento, este será un vértice exterior. Si la clase tiene más de un elemento, entonces todos estos serán vértices interiores.

Para construir el conjunto de vértices del grafo reducido, se crea un vértice por cada clase de P.

Para adicionar una arista al grafo reducido, los dos vértices que la forman deben pertenecer a clases distintas. En caso de que exista más de una arista, en el grafo de entrada del algoritmo, entre los vértices que pertenecen a dos clases distintas, se adiciona la de menor costo.

**(técnicas a utilizar)**

* Con el algoritmo de Dijkstra es posible determinar la distancia más corta (de menos esfuerzo o costo) entre un nodo inicial  y cualquier otro nodo en un grafo. La idea del algoritmo es calcular continuamente la distancia más corta desde un punto inicial y excluyendo las distancias más largas cuando se efectúa una actualización. Como ejemplo, en el siguiente *grid*, se asume que los nodos en morado son sumamente costosos, por lo que el algoritmo de Dijkstra deberá hallar la ruta que minimice el costo total entre cualquier tupla de nodos, como por ejemplo, pasar del nodo "A" al nodo "W"



El algoritmo consiste formalmente en los siguientes pasos:

1. Inicialización de todos los nodos con distancia "infinita", iniciar el nodo de comienzo con 0.
2. Marcar las distancias del nodo inicial como permanentes, todas las demás como temporales.
3. Establecer el nodo inicial como activo.
4. Cálculo de las distancias temporales de todos los nodos vecinos a el nodo activo sumando sus distancias con los pesos de los arcos.
5. Si la distancia calculada de un nodo es menor a la actual, actualiza la distancia y fija el nodo actual como antecesor, Este paso es llamado actualización y es la idea central de Dijkstra.
6. Estableciendo el nodo con la distancia temporal mínima como activa, marca sus distancia como permanente.
7. Repita los pasos del 4 al 7 hasta que no queden nodos con una distancia permanente cuyos vecinos tengan aún distancias temporales.

**(ultilidad en la vida real)**

* El algoritmo de Dijkstra se aplica sobre un grafo ponderado y calcula la distancia desde un vértice inicial al resto de vértices del grafo. Los GPS tratan el mapa de carreteras como un grafo ponderado (cuyos pesos son o bien la distancia o el tiempo) y utilizan este algoritmo para calcular el camino más corto entre dos puntos.

<https://jariasf.wordpress.com/2012/03/19/camino-mas-corto-algoritmo-de-dijkstra/>

<http://instintologico.com/dijkstra-y-el-camino-mas-corto/>

<https://alexplatasl.blogspot.com/2017/11/algoritmo-de-dijkstra-en-r.html>