**Algoritmo de Dijkstra**

Fue creado por Edger W. Dijkstra en 1956, sirve para cualquier grafo con peso (sea dirigido o no) siempre y cuando sus pesos no sean negativos. Este algoritmo tiene como idea:

* Calcular distancias mínimas desde un nodo inicial a todos los demás.

Para poder hacer esto, en cada paso se debe tomar el nodo más cercano al inicial que aún no fue visitado. Después recalcularnos todos los caminos mínimos teniendo en cuenta a los que no fueron visitados como un camino intermedio, luego en cada paso tendremos un subconjunto de nodos que ya tiene calculada su distancia mínima y los demás tienen calculada su mínima distancia si solo puedo usar los nodos del conjunto como nodos intermedios y para finalizar con cada iteración agregamos un nodo mas a nuestro conjunto hasta resolver el problema en su totalidad.

Ejemplo del código sin cola de prioridad:

Dijkstra (Grafo G, nodo inicial s)

visitado[n] = {false , ... , false} // guarda si un nodo ya fue visitado

distancia[n] = { Infinito , ... , Infinito } // guarda las distancias del nodo salida al resto

para cada w en V[G] hacer

si existe arista entre s y w entonces

distancia[w] = peso (s, w)

distancia[s] = 0

visitado[s] = true

mientras que no esten visitados todos hacer

v = nodo de menor distancia a s que no fue visitado aun

visitado[v] = true

para cada w en sucesores (G, v) hacer

si distancia[w] > distancia[v] + peso (v, w) entonces

distancia[w] = distancia[v] + peso (v, w)

padre[w] = v

Ejemplo del código con cola de prioridad:

Dijkstra (Grafo G, nodo\_fuente s)

para todo u en V[G] hacer

distancia[u] = INFINITO

padre[u] = NULL

visitado[u] = false

distancia[s] = 0

adicionar (cola, (s, distancia[s] ) )

mientras que cola no sea vacia hacer

u = extraer\_minimo(cola)

visitado[u] = true

para todo v en adyacencia[u] hacer

si no visitado[v] y distancia[v] > distancia[u] + peso (u, v) hacer

distancia[v] = distancia[u] + peso (u, v)

padre[v] = u

adicionar (cola, ( v, distancia[v] ) )

**Algoritmo de a\* (a estrella)**

A\* es un algoritmo de búsqueda inteligente o informada que busca el camino más corto desde un estado inicial al estado meta a través de un espacio de problema, usando una heurística óptima. Como ignora los pasos más cortos (más "chatos") en algunos casos rinde una solución subóptima.

PSEUDOCÓDIGO DE A\*:

Inicializar lista ABIERTA

Inicializar lista CERRADA

Crear nodo objetivo; llamarlo nodo\_meta

Crear nodo inicial; llamarlo nodo\_inicio

Agregar nodo\_inicio a la lista ABIERTA

Mientras la lista ABIERTA no esté vacía

{

Obtener el nodo n de la lista ABIERTA con el valor f(n) más bajo

Agregar n a la lista CERRADA

Si n es igual a nodo\_meta hemos encontrado la solución; devolver Solución(n)

Generar cada nodo sucesor n' de n

Para cada nodo sucesor n' de n

{

Establecer el padre de n' como n

Establecer h(n') como la estimación heurística de la distancia a nodo\_meta

Establecer g(n') como g(n) más el costo para llegar a n' desde n

Establecer f(n') como g(n') más h(n')

Si n' está en la lista ABIERTA y el existente es igual o mejor, descartar n' y continuar

Si n' está en la lista CERRADA y el existente es igual o mejor, descartar n' y continuar

Eliminar ocurrencias de n' de ABIERTA y CERRADA

Agregar n' a la lista ABIERTA

}

}

Devolver fallo (si llegamos a este punto hemos buscado todos los nodos alcanzables y aún no hemos encontrado la solución, por lo tanto, una no existe)