Dijsktra

Matemáticas computacionales

Yarethzi Giselle Bazaldúa Parga

20 de octubre de 2017

Resumen

El algoritmo de Dijsktra, también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de los vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

Disjktra

Como ya se mencionó, el fin de éste algoritmo es encontrar el camino más corto desde un vértice determinado al que llamaremos “origen” a cada uno de los demás vértices con peso en cada arista, la complejidad del algoritmo es de O(n^2). Éste algoritmo se realizó con ayuda del maestro en el salón de clases, además de tomar como base otro algoritmo de grafos previamente hecho. El pseudocódigo es el siguiente:

```
>>> class Grafo:
        def __init__(self):
                self.V = set()
                self.E = dict()
                self.vecinos = dict()
        def agrega(self, v):
                self.V.add(v)
                if not v in self.vecinos:
                        self.vecinos[v] = set()
        def conecta(self, v, u, peso=1):
                self.agrega(v)
                self.agrega(u)
                self.E[(v,u)] = self.E[(u,v)] = peso
                self.vecinos[v].add(u)
                self.vecinos[u].add(v)
        def complemento(self):
                comp= Grafo()
                for v in self.V:
                        for w in self.V:
                                if v != w and (v,w) not in self.E:
                                        comp.conecta(v, w, 1)
                return comp

        def shortest(self, v, w):
                        q = [(0, v, ())]
                        visited = set()
                        while len(q) > 0:
                                (l, u, p) = heappop(q)
                                if u not in visited:
                                        visited.add(u)
                                        if u == w:
                                                return list(flatten(p))[::-1] + [
                                p = (u, p)
                                for n in self[u].neighbors:
                                        if n not in visited:
                                                el = self.vecinos[u][n]
                                                heappush(q, (l + el, n, p))
                        return None
```