**Bellman-Ford**

Funcionamiento: Este algoritmo se basa en la propiedad que indica que, en un grafo no dirigido sin ciclos negativos, un camino mínimo entre dos nodos cualesquiera tiene a lo sumo |V| pasos. Luego, el algoritmo realiza |V| iteraciones, en las que recorre todas las aristas en cada una de ellas. En la iteración sobre la arista (u, v), chequea qué es mejor: “*llegar a v con el camino que ya encontré o pasando por u, sumando el peso de la arista”*. Con esto, se realizan |V|·|E| iteraciones, siendo esa la complejidad del algoritmo.

Este algoritmo corresponse a la programación dinámica, ya que *tiene memoria en cada iteración sobre el coste de camino en iteraciones anteriores*.

Ventajas: Resuelve el problema siempre que no haya ciclos negativos. Detecta los ciclos negativos.

Desventajas: La complejidad es O(|V|·|E|). No encuentra el camino que no repita ningún vértice, en caso de que existan ciclos negativos.

Detección de ciclos negativos: Basándonos en la propiedad indicada al comienzo, una vez que el algoritmo termina las |V| iteraciones sobre cada arista, se deberían conocer todos los caminos mínimos. Por esto, se pueden recorrer todas las aristas una vez más. De esta forma, si se encuentra un mejor camino significa que se viola la propiedad, por lo que se puede concluir que en dicho grafo existe un ciclo negativo.

Complejidad: Como se vio anteriormente, la complejidad del algoritmo es O(|V|·|E|). Pero, en nuestro problema de grafos completos, |E| = |V|2-|V|, por lo que la complejidad queda O(|V|·(|V|2-|V|))=O(|V|3)

En el siguiente gráfico se puede ver la evolución de los tiempos para casos de 100, 200, 300, 400, 500 y 1000 aristas.

