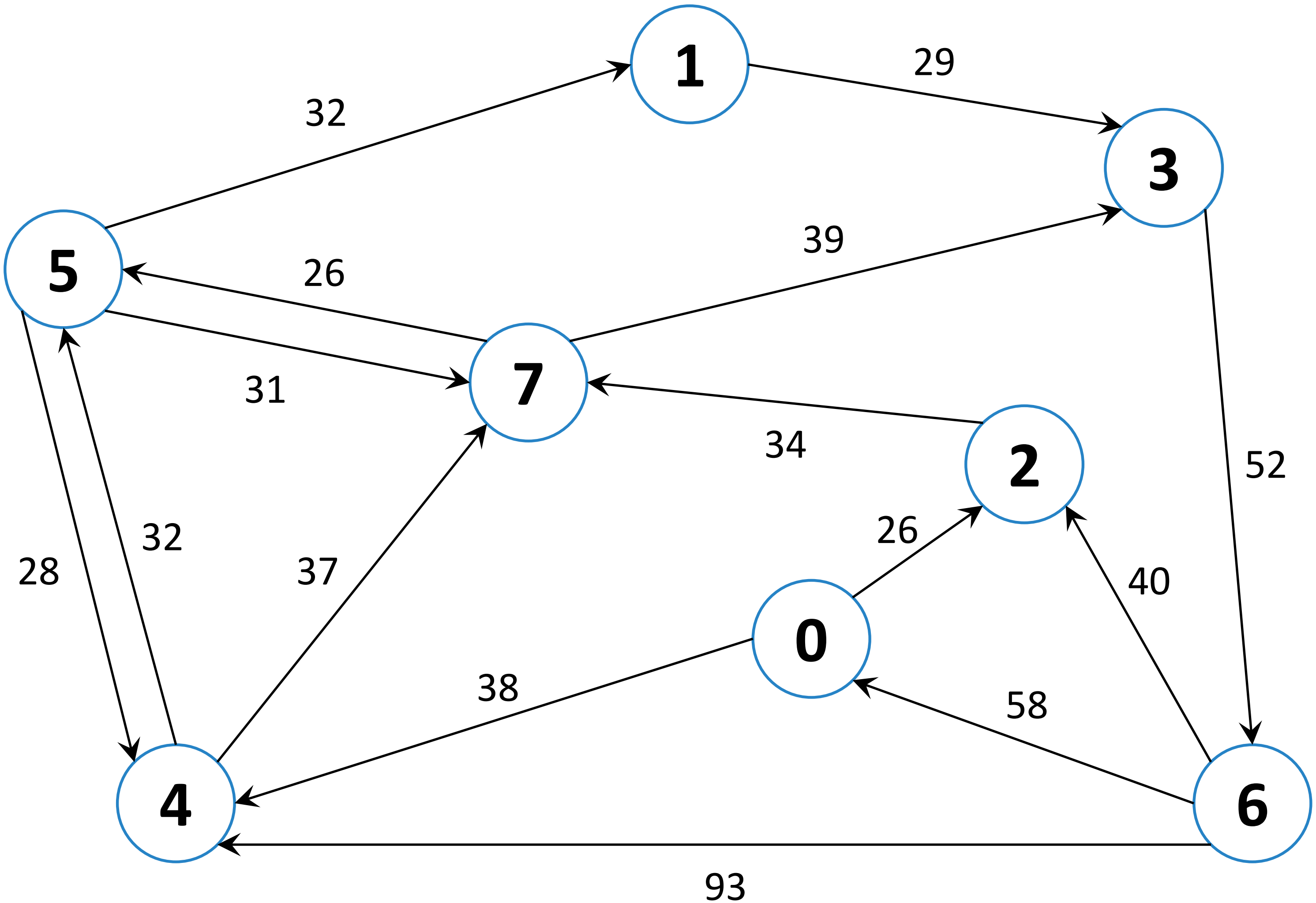
El viaje familiar

- Quieres planificar un viaje en auto desde A a B
- El rendimiento de tu auto depende de la velocidad a la que vaya
- Algunos caminos tienen peaje
- Cada camino tiene su propia velocidad obligatoria

¿Cómo hacer para que el viaje te salga lo más barato posible?

¿Y lo más corto posible?

Grafo direccional con costos



La ruta más barata (o la más corta)

Debemos buscar la **ruta más barata** de A a B, es decir,

La **suma de los costos** de sus aristas debe ser mínima

¿Podremos aprovechar un algoritmo que conozcamos?

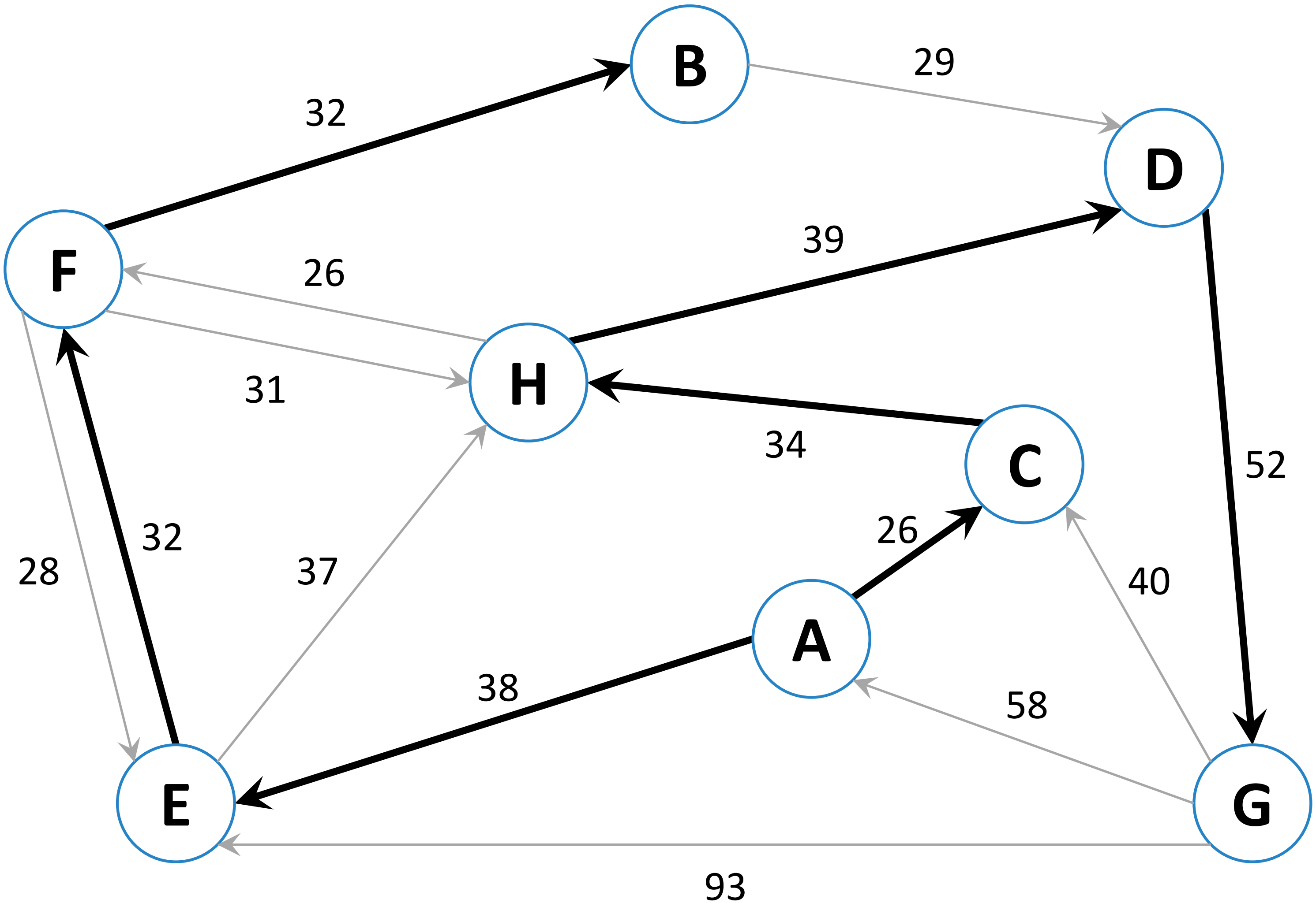
Propiedades de BFS

¿Cuál es la propiedad que garantiza la corrección de **BFS**?

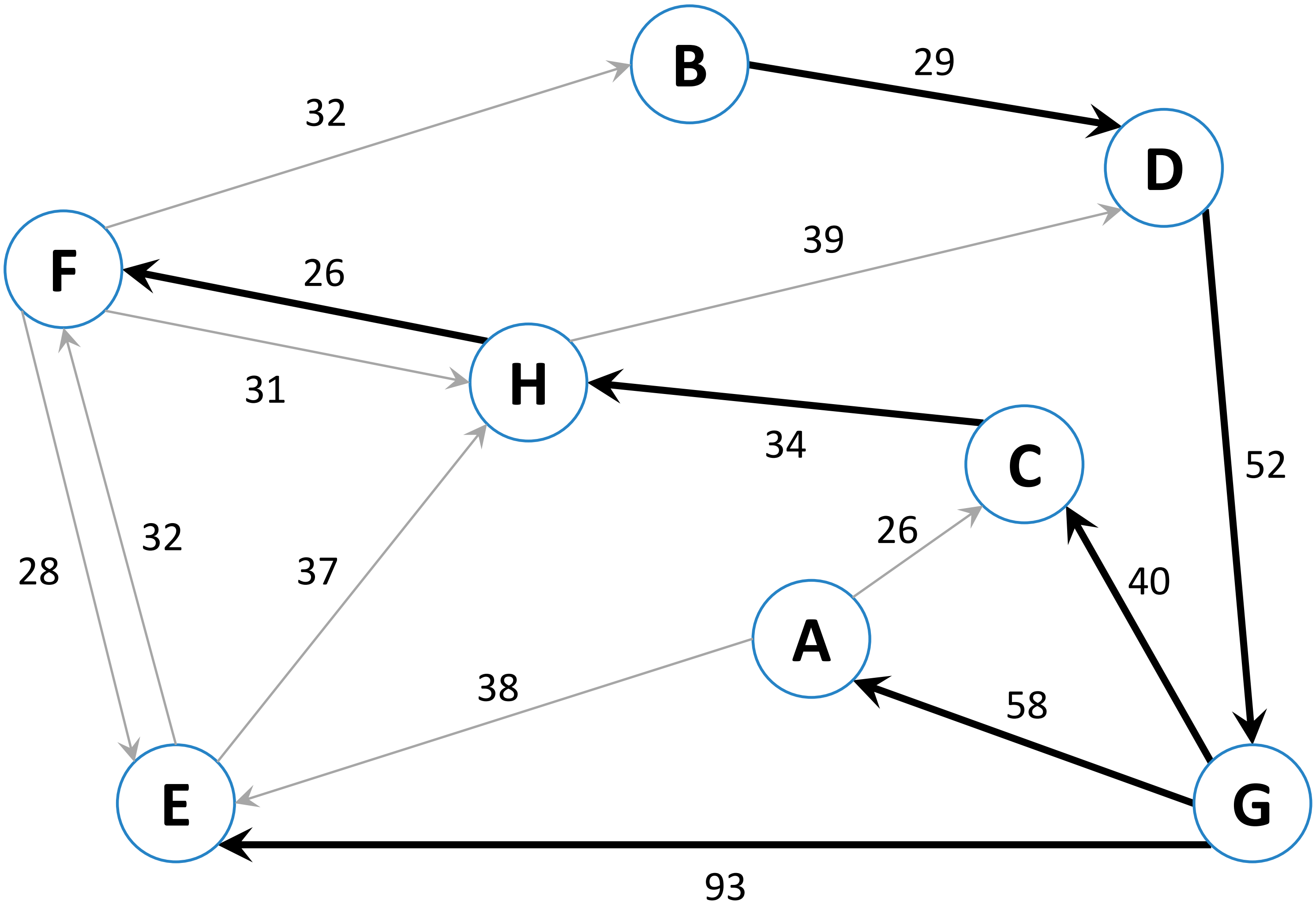
Si **marcamos** cada arista que queda como padre de un nodo,

¿Qué **representa** el conjunto de aristas marcadas?

Árbol de rutas más cortas desde *A*



Árbol de rutas más cortas desde *B*



Propiedades del problema de rutas más cortas

Las rutas son direccionales

Los costos no son solo distancias

Puede que haya vértices inalcanzables desde el vértice de partida

Si hay costos negativos, es más complicado resolver el problema

Las rutas más cortas pueden no ser únicas

BFS++

¿Cómo podemos extender **BFS** para este problema?

Queremos **garantizar** que al sacar un nodo de **Open**, hemos encontrado a ese nodo por la ruta más corta

$\boldsymbol{bfs}(\boldsymbol{G}(\boldsymbol{V},\boldsymbol{E}),\boldsymbol{s},\boldsymbol{g})$:

$\boldsymbol{s.dist} \leftarrow \boldsymbol{0}$

$\boldsymbol{Open} \leftarrow$ una cola conteniendo únicamente a $\boldsymbol{s}$

Insertar $\boldsymbol{s}$ en $\boldsymbol{Open}$

$\boldsymbol{while}$ $\boldsymbol{Open} \neq \emptyset$:

$\boldsymbol{u} \leftarrow$ extraer y pintar el siguiente elemento de $\boldsymbol{Open}$

$\boldsymbol{if}$ $\boldsymbol{u} = \boldsymbol{g}$, $\boldsymbol{return\ true}$

$\boldsymbol{foreach}$ $(\boldsymbol{u},\boldsymbol{v}) \in \boldsymbol{E}$:

$\boldsymbol{if}$ $\boldsymbol{v}$ está pintado, $\boldsymbol{continue}$

$\boldsymbol{d} \leftarrow \boldsymbol{u.dist} + \boldsymbol{1}$

$\boldsymbol{if}$ $\boldsymbol{d} \geq \boldsymbol{v.dist}$, $\boldsymbol{continue}$

$\boldsymbol{v.parent} \leftarrow \boldsymbol{u}, \quad \boldsymbol{v.dist} \leftarrow \boldsymbol{d}$

Insertar $\boldsymbol{v}$ en $\boldsymbol{Open}$

$\boldsymbol{return\ false}$

$\boldsymbol{bfs++(G(V,E),s,g)}$:

$\boldsymbol{s.dist \leftarrow 0}$

$\boldsymbol{Open} \leftarrow$ una cola de prioridades conteniendo únicamente a $\boldsymbol{s}$

Insertar $\boldsymbol{s}$ en $\boldsymbol{Open}$ con prioridad $\boldsymbol{s.dist}$

$\boldsymbol{while\ Open \neq \emptyset}$:

$\boldsymbol{u} \leftarrow$ extraer y pintar el siguiente elemento de $\boldsymbol{Open}$

$\boldsymbol{if\ u = g,\ return\ true}$

$\boldsymbol{foreach\ (u,v) \in E}$:

$\boldsymbol{if\ v}$ está pintado, $\boldsymbol{continue}$

$\boldsymbol{d \leftarrow u.dist + w(u,v)}$

$\boldsymbol{if\ d \geq v.dist,\ continue}$

$\boldsymbol{v.parent \leftarrow u, \quad v.dist \leftarrow d}$

Insertar o actualizar $\boldsymbol{v}$ en $\boldsymbol{Open}$ con prioridad $\boldsymbol{v.dist}$

$\boldsymbol{return\ false}$

Algoritmo de Dijkstra

Este algoritmo se conoce como el algoritmo de **Dijkstra**

Parece tener sentido, pero, ¿es **correcto**?

¿Se está cumpliendo la **garantía** que propusimos?

$\boldsymbol{dijkstra}(\boldsymbol{G}(\boldsymbol{V},\boldsymbol{E}),\boldsymbol{s},\boldsymbol{g})$:

$\boldsymbol{s.dist} \leftarrow \mathbf{0}$

$\boldsymbol{Open} \leftarrow$ una cola de prioridades conteniendo únicamente a $\boldsymbol{s}$

Insertar $\boldsymbol{s}$ en $\boldsymbol{Open}$ con prioridad $\boldsymbol{s.dist}$

$\boldsymbol{while}$ $\boldsymbol{Open} \neq \emptyset$:

$\boldsymbol{u} \leftarrow$ extraer y pintar el siguiente elemento de $\boldsymbol{Open}$

$\boldsymbol{if}$ $\boldsymbol{u} = \boldsymbol{g}$, $\boldsymbol{return\ true}$

$\boldsymbol{foreach}$ $(\boldsymbol{u},\boldsymbol{v}) \in \boldsymbol{E}$:

$\boldsymbol{if}$ $\boldsymbol{v}$ está pintado, $\boldsymbol{continue}$

$\boldsymbol{d} \leftarrow \boldsymbol{u.dist} + \boldsymbol{w}(\boldsymbol{u},\boldsymbol{v})$

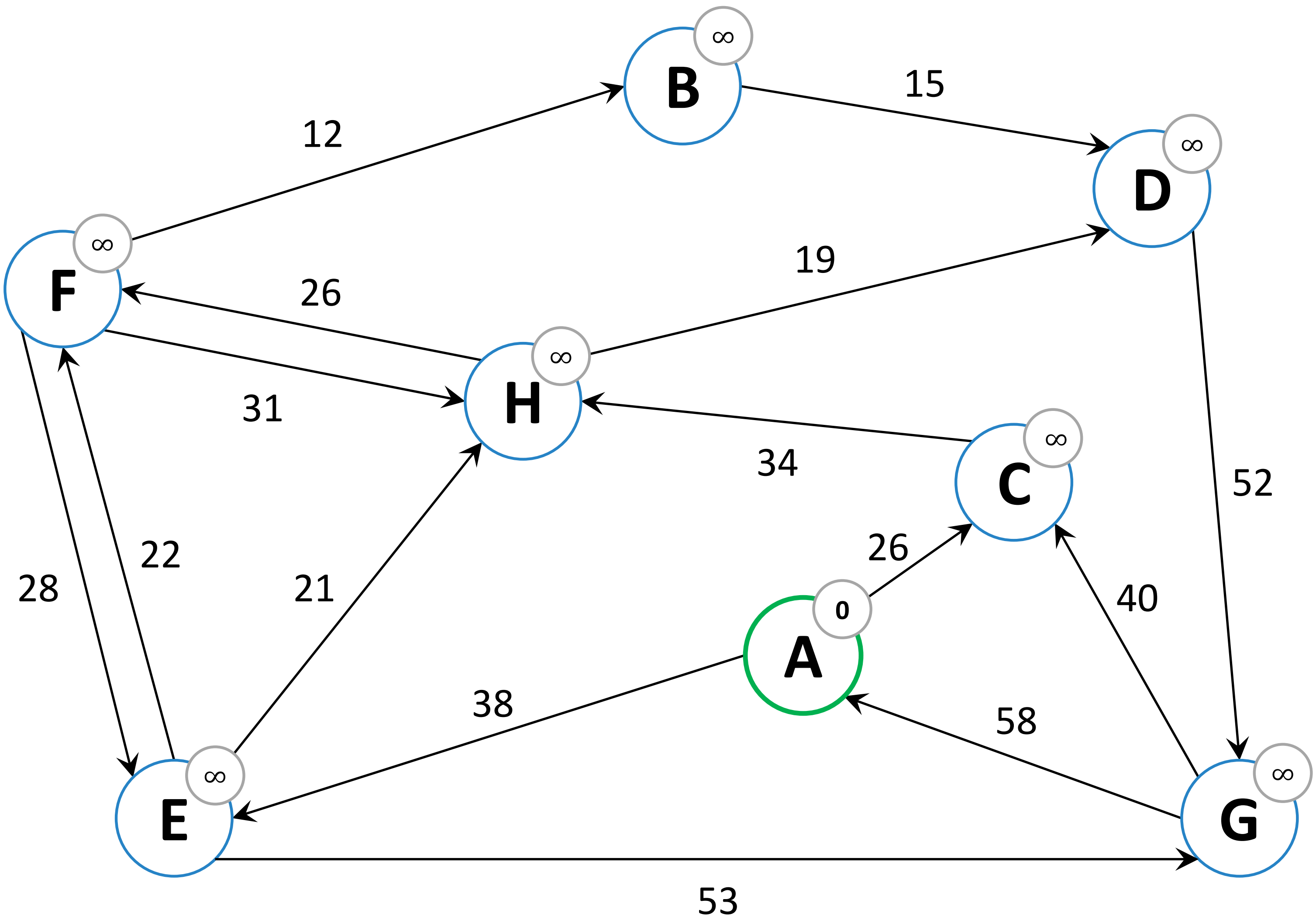
$\boldsymbol{if}$ $\boldsymbol{d} \geq \boldsymbol{v.dist}$, $\boldsymbol{continue}$

$\boldsymbol{v.parent} \leftarrow \boldsymbol{u}, \quad \boldsymbol{v.dist} \leftarrow \boldsymbol{d}$

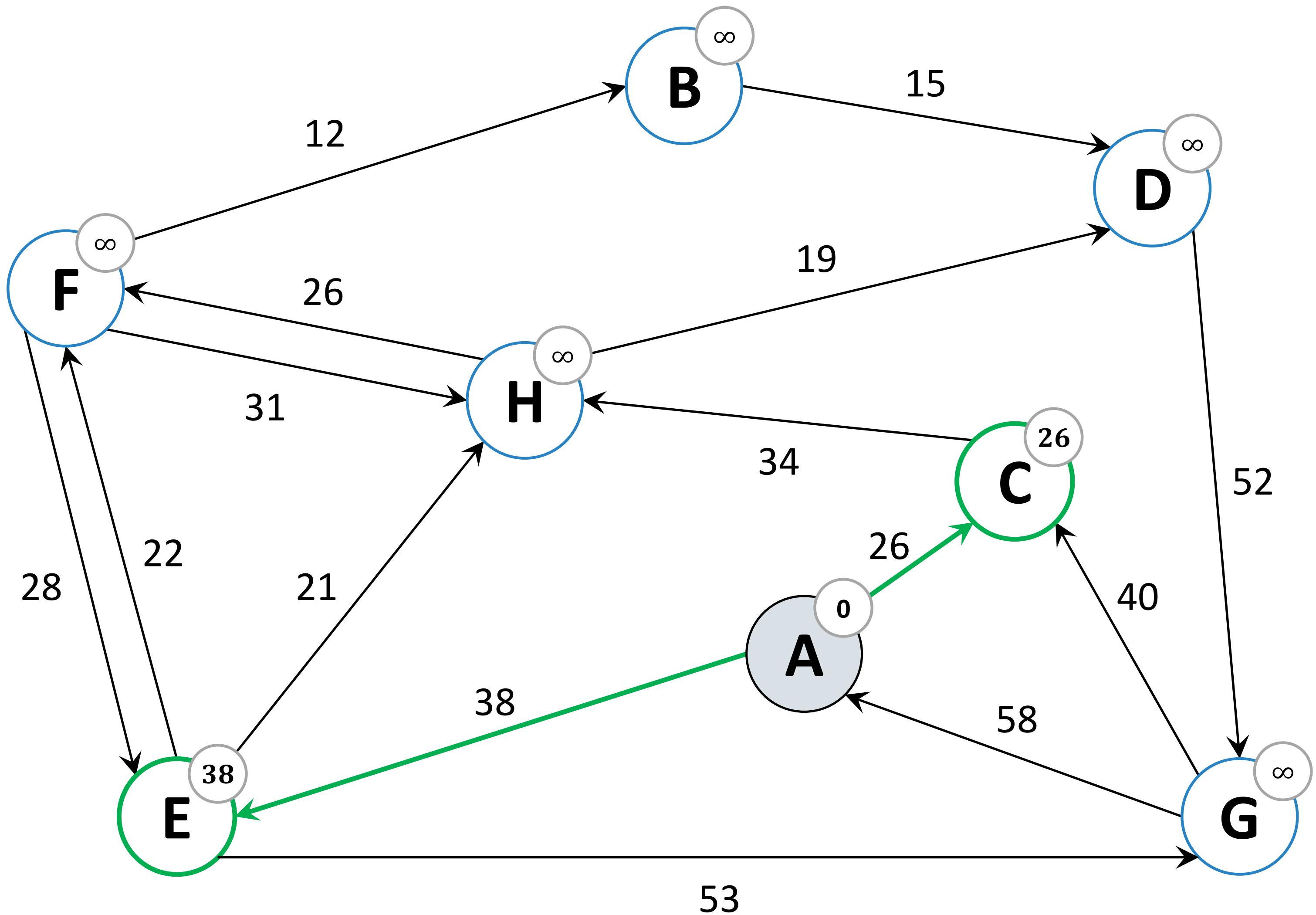
Insertar o actualizar $\boldsymbol{v}$ en $\boldsymbol{Open}$ con prioridad $\boldsymbol{v.dist}$

$\boldsymbol{return\ false}$

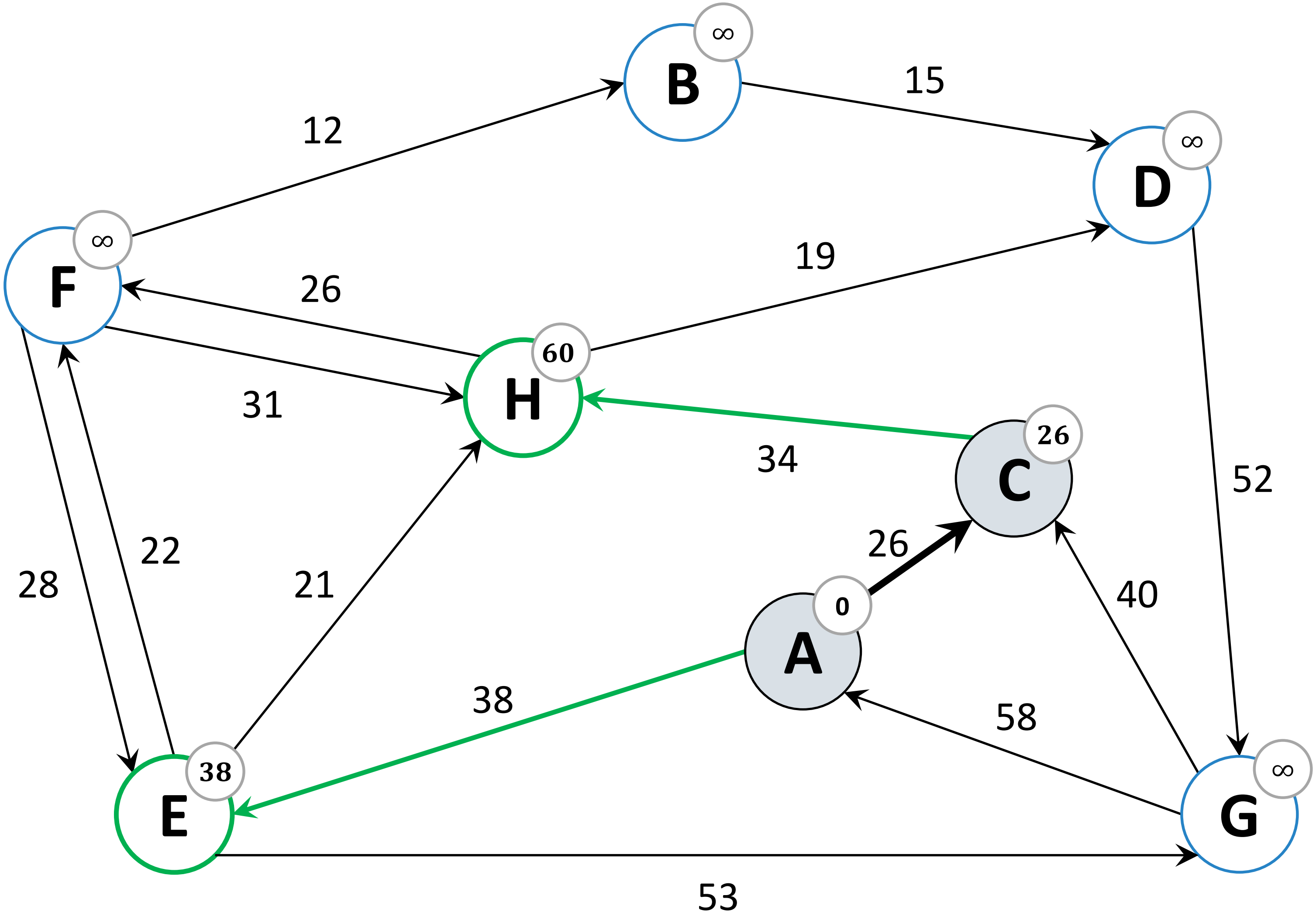
El algoritmo de Dijkstra en acción



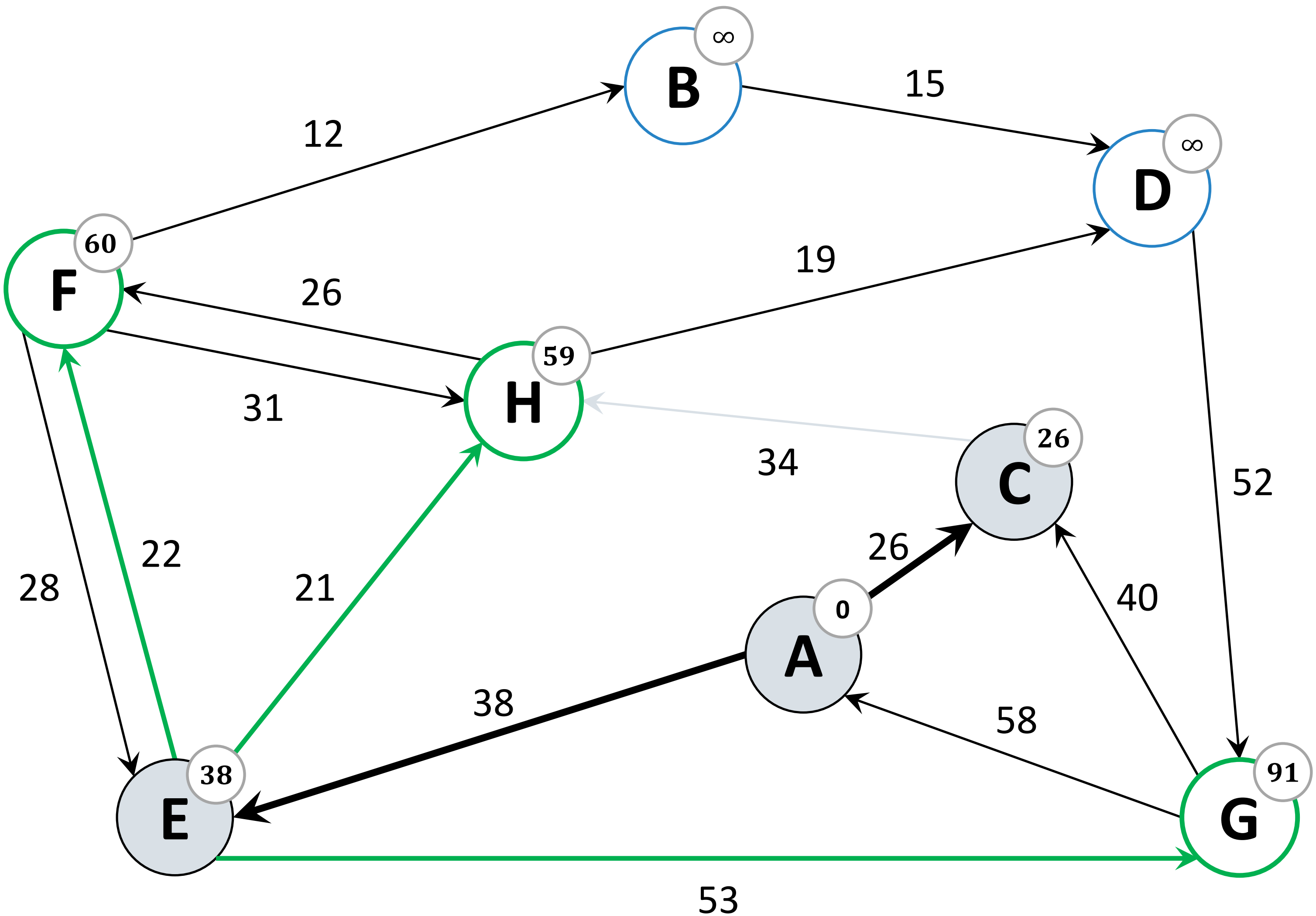
El algoritmo de Dijkstra en acción



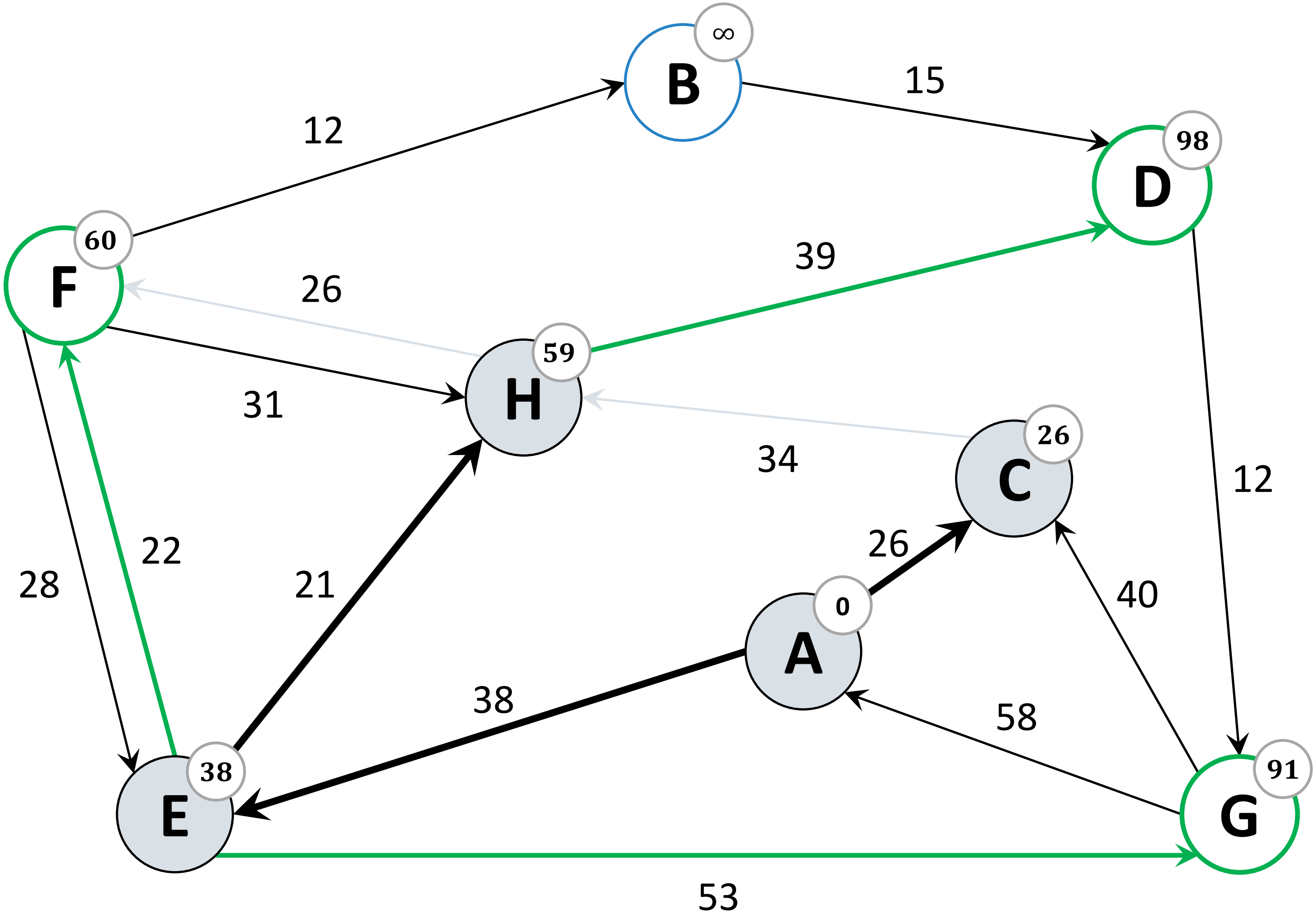
El algoritmo de Dijkstra en acción



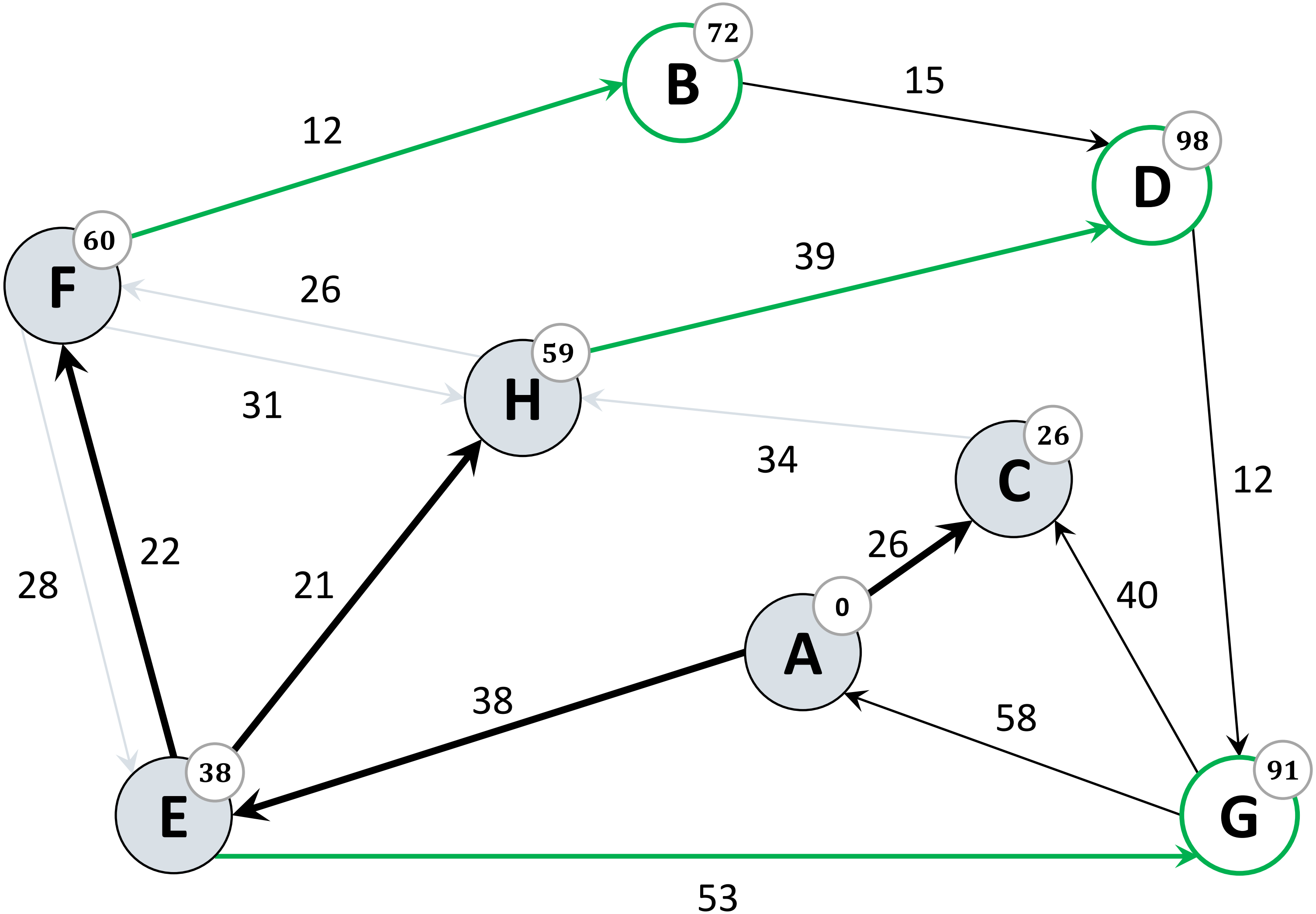
El algoritmo de Dijkstra en acción



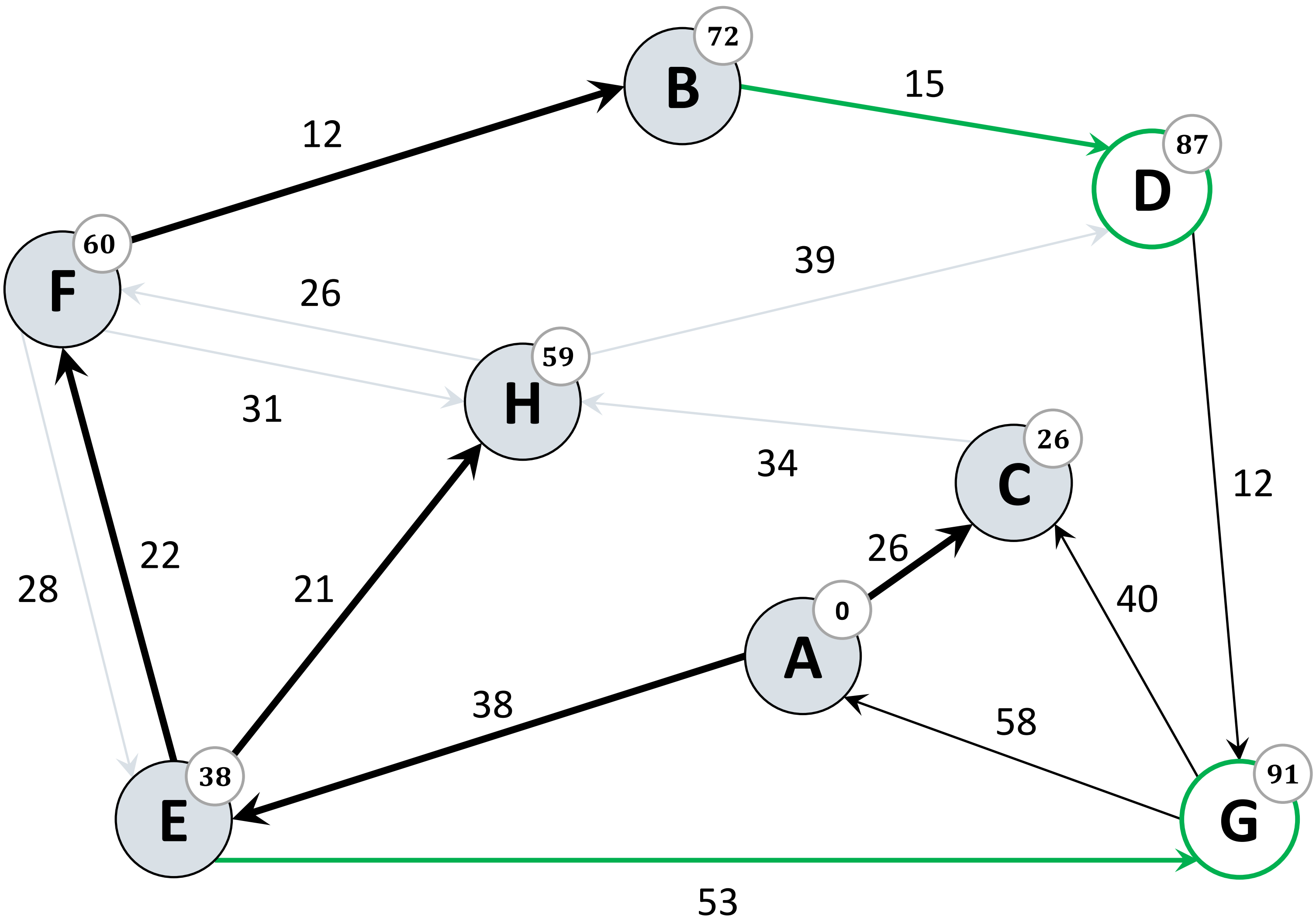
El algoritmo de Dijkstra en acción



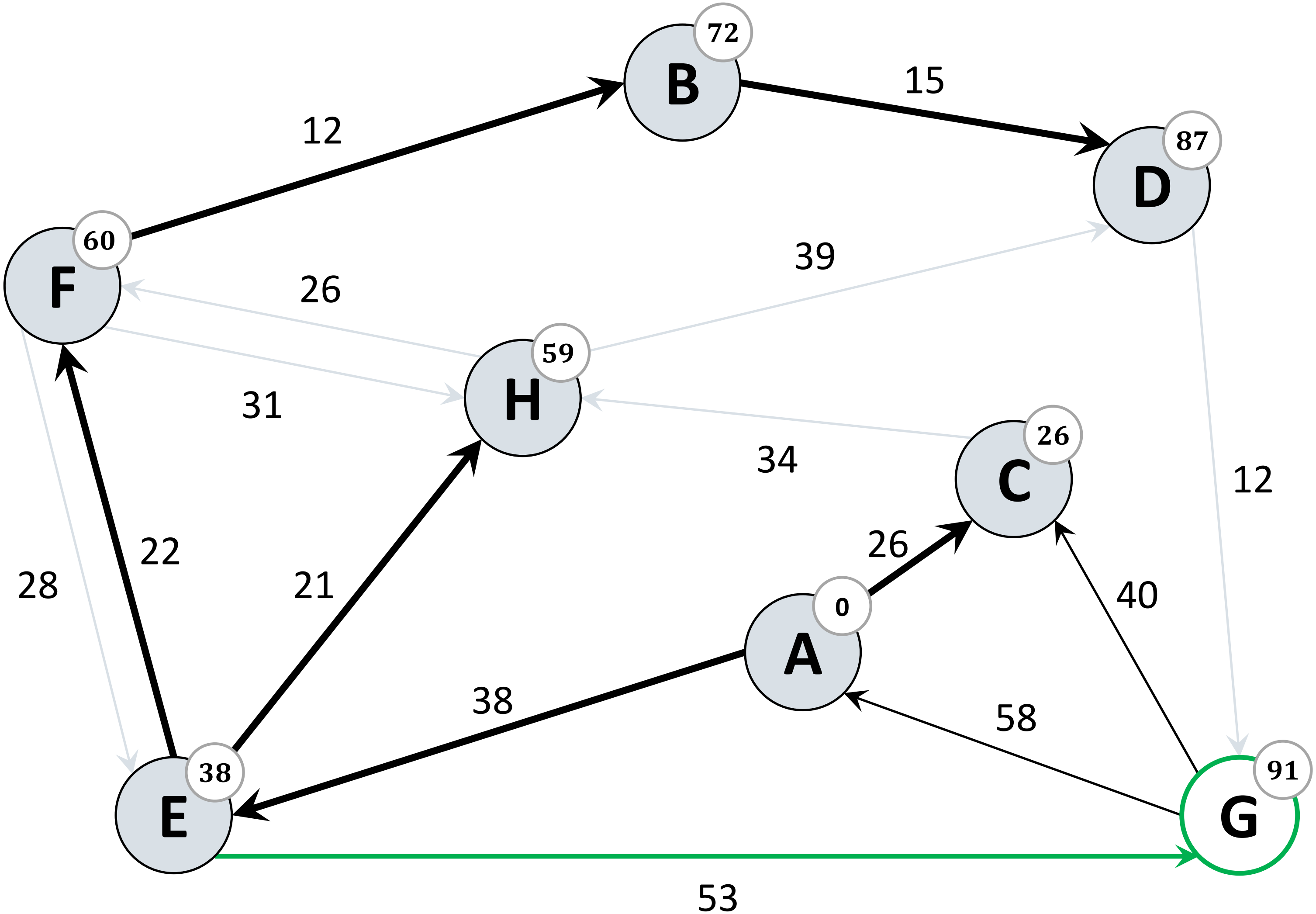
El algoritmo de Dijkstra en acción



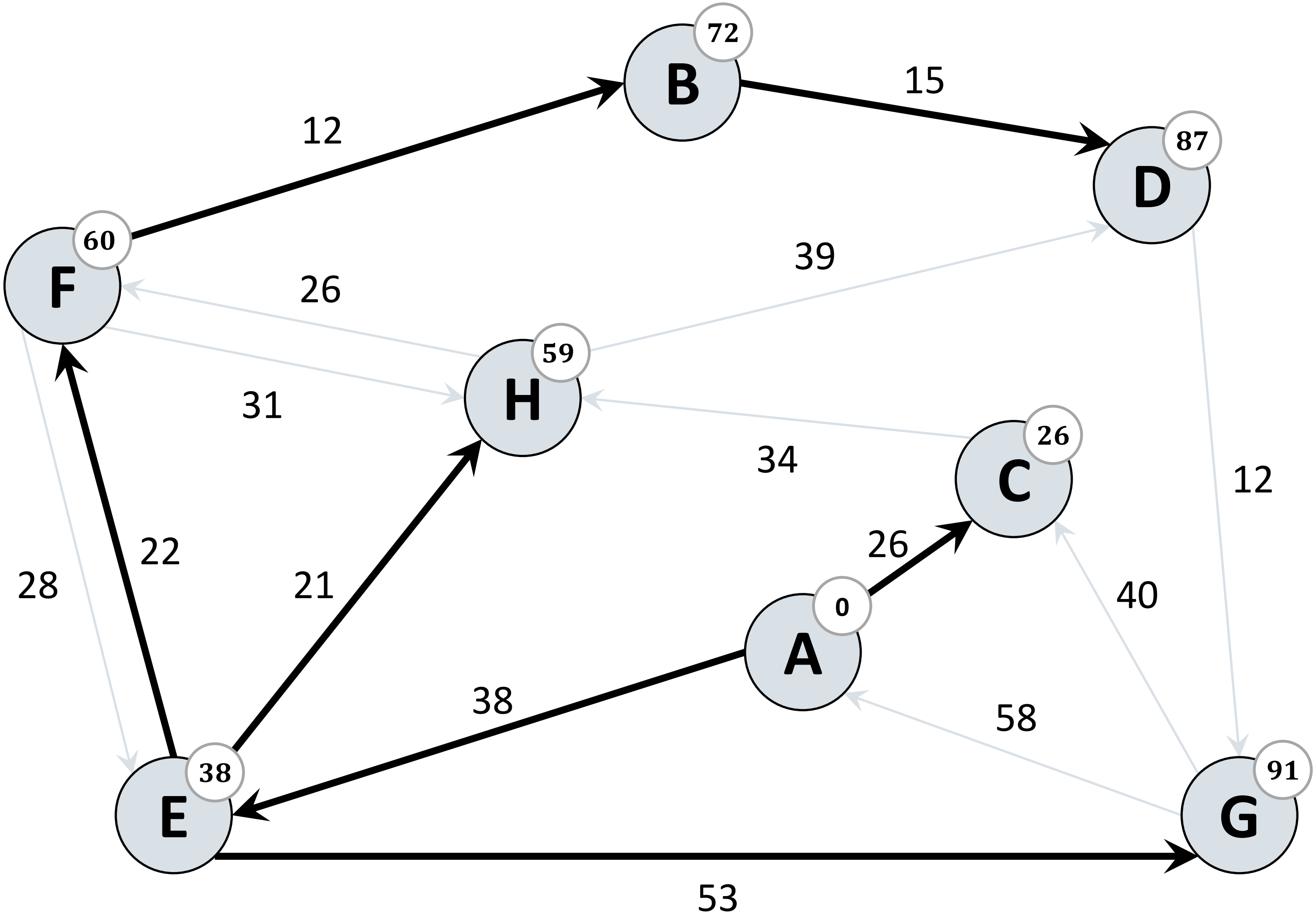
El algoritmo de Dijkstra en acción



El algoritmo de Dijkstra en acción



El algoritmo de Dijkstra en acción



Complejidad

¿Cuál es la complejidad del algoritmo de Dijkstra?

Algoritmos codiciosos

El algoritmo de **Dijkstra** es **codicioso**

Estos algoritmos no necesariamente producen soluciones **óptimas**

¿Por qué funciona el enfoque codicioso en este caso?

Variantes

Rutas más cortas en grafos acíclicos

Rutas más cortas de un vértice a otro

Rutas más cortas entre todos los pares de vértices

Rutas más cortas en grafos euclideanos