Shortest and longest paths on DAGs

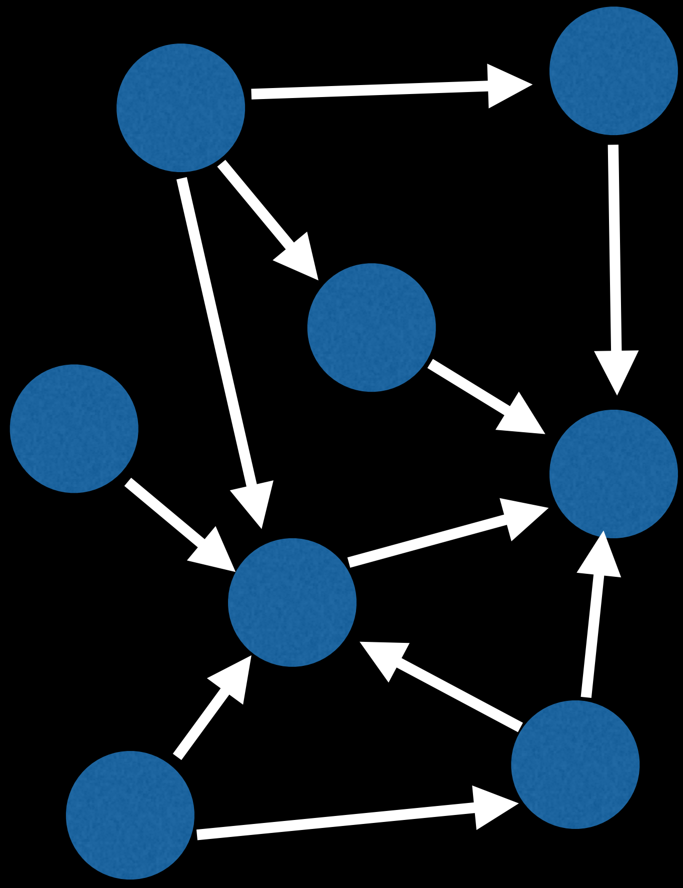
William Fiset

Directed Acyclic Graph (DAG)

Recall that a **Directed Acyclic Graph (DAG)** is a graph with directed edges and no cycles. By definition this means all **trees** are automatically DAGs since they do not contain cycles.

Directed Acyclic Graph (DAG)

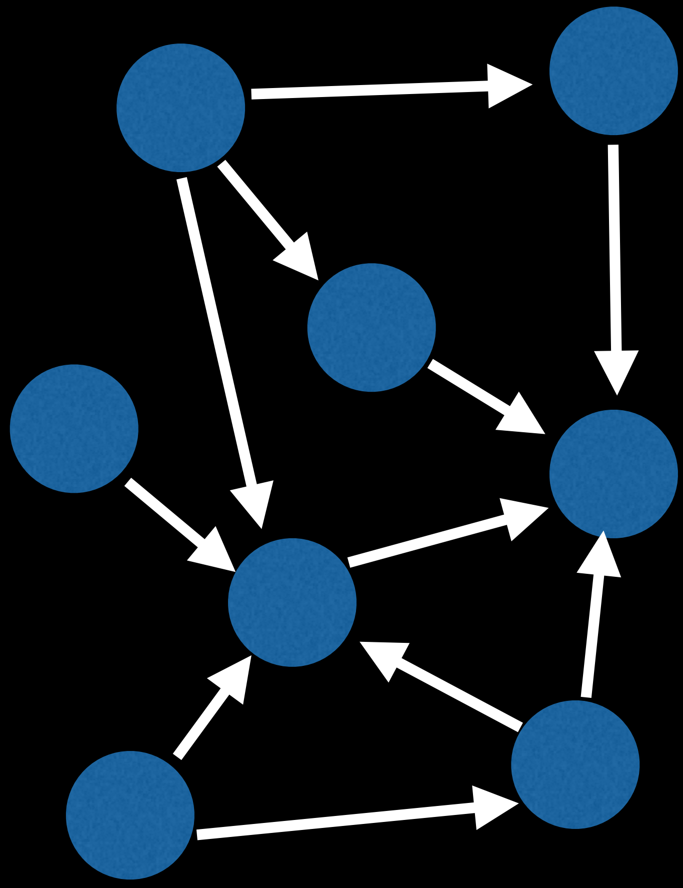
Recall that a **Directed Acyclic Graph (DAG)** is a graph with directed edges and no cycles. By definition this means all **trees** are automatically DAGs since they do not contain cycles.



Q: Is this graph a DAG?

Directed Acyclic Graph (DAG)

Recall that a **Directed Acyclic Graph (DAG)** is a graph with directed edges and no cycles. By definition this means all **trees** are automatically DAGs since they do not contain cycles.

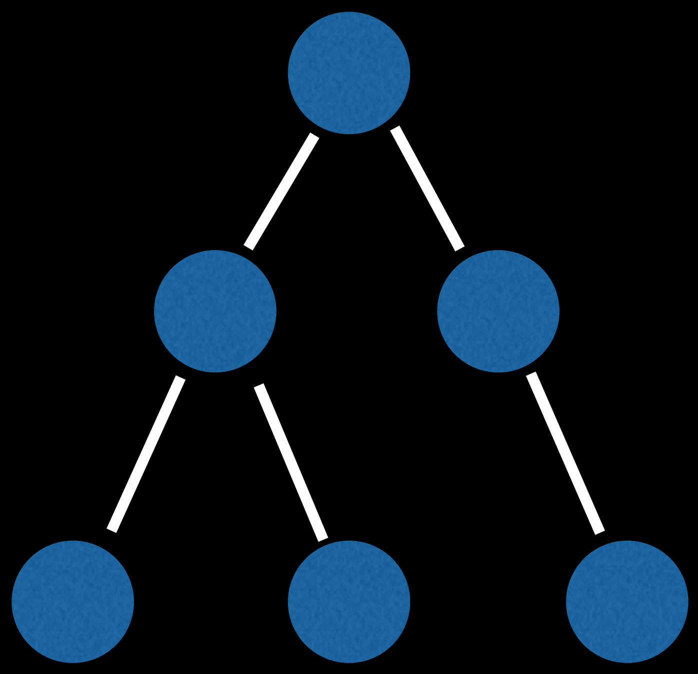


Q: Is this graph a DAG?

A: Yes!

Directed Acyclic Graph (DAG)

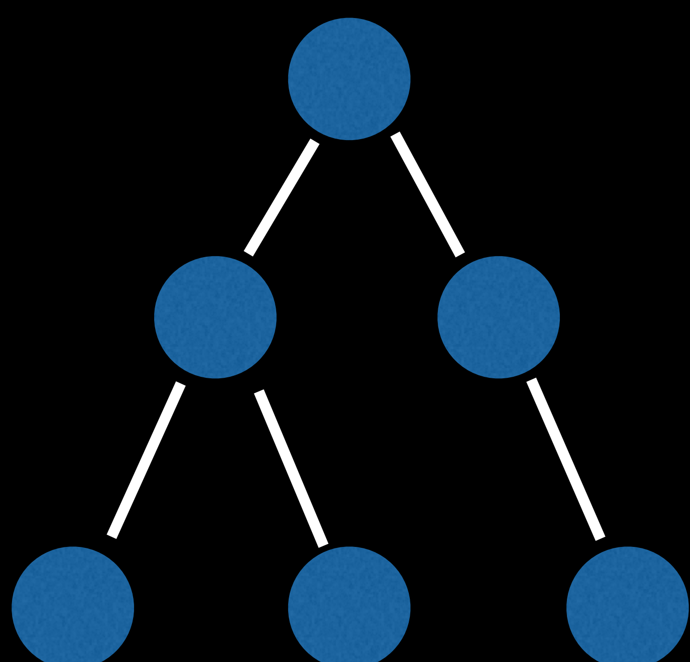
Recall that a **Directed Acyclic Graph (DAG)** is a graph with directed edges and no cycles. By definition this means all **trees** are automatically DAGs since they do not contain cycles.



Q: Is this graph a DAG?

Directed Acyclic Graph (DAG)

Recall that a **Directed Acyclic Graph (DAG)** is a graph with directed edges and no cycles. By definition this means all **trees** are automatically DAGs since they do not contain cycles.



Q: Is this graph a DAG?

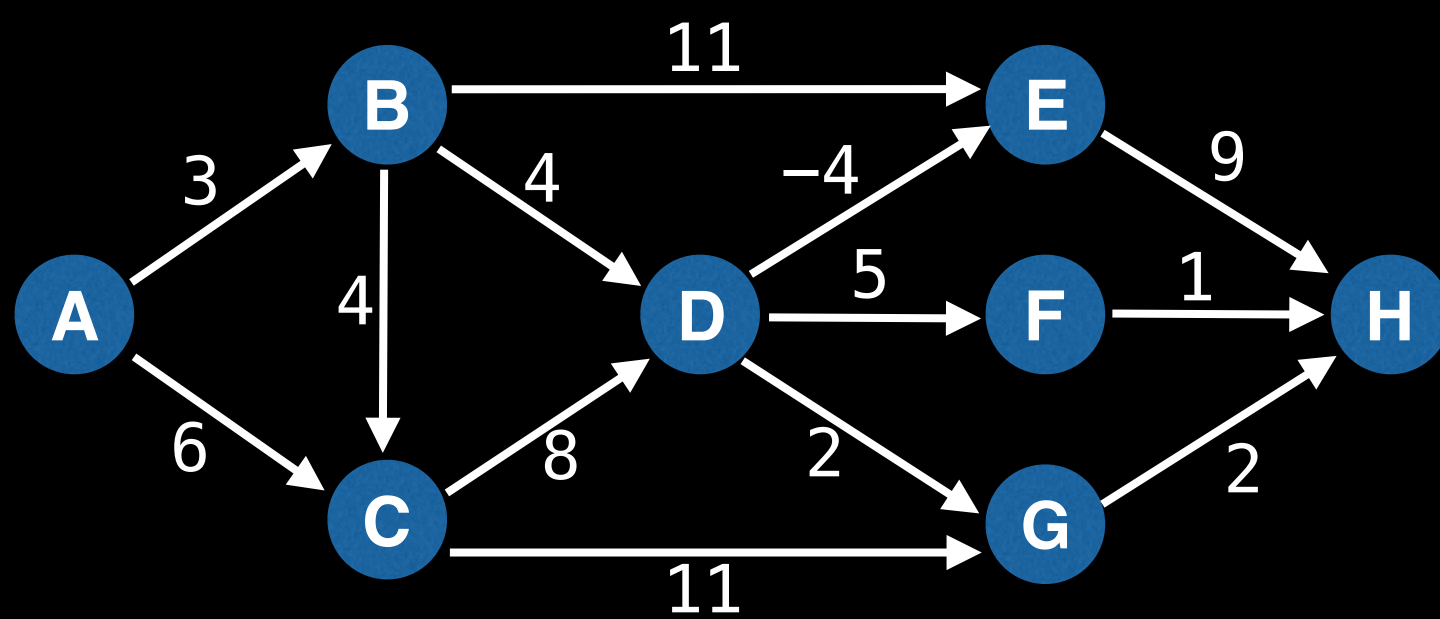
A: No, the structure may be a tree, but it does not have directed edges.

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

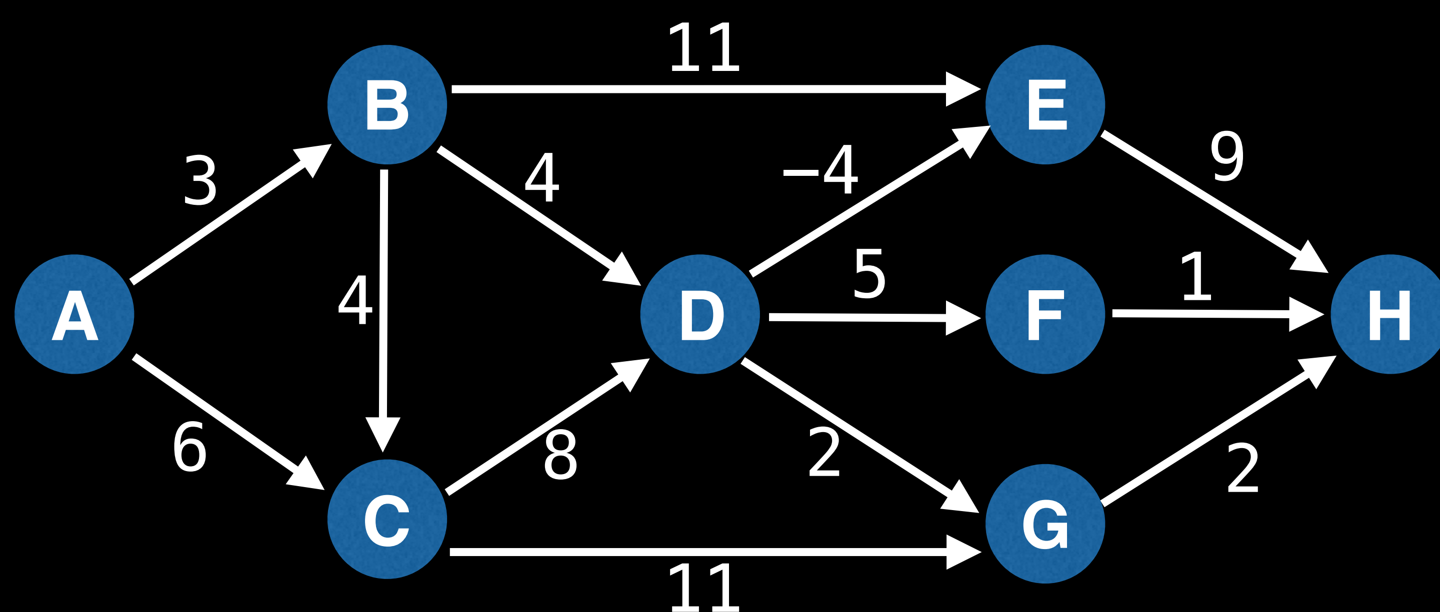
SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.



SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

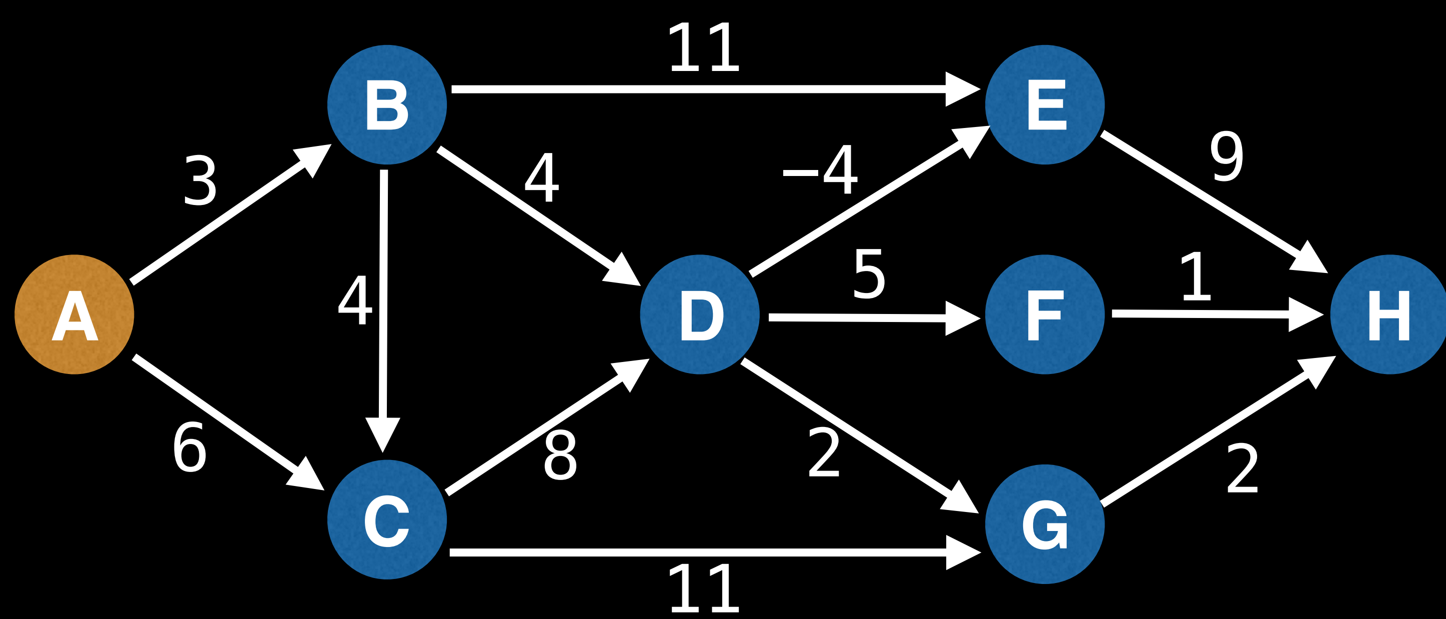


Arbitrary topological order: A, B, C, D, G, E, F, H

∞	∞	∞	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

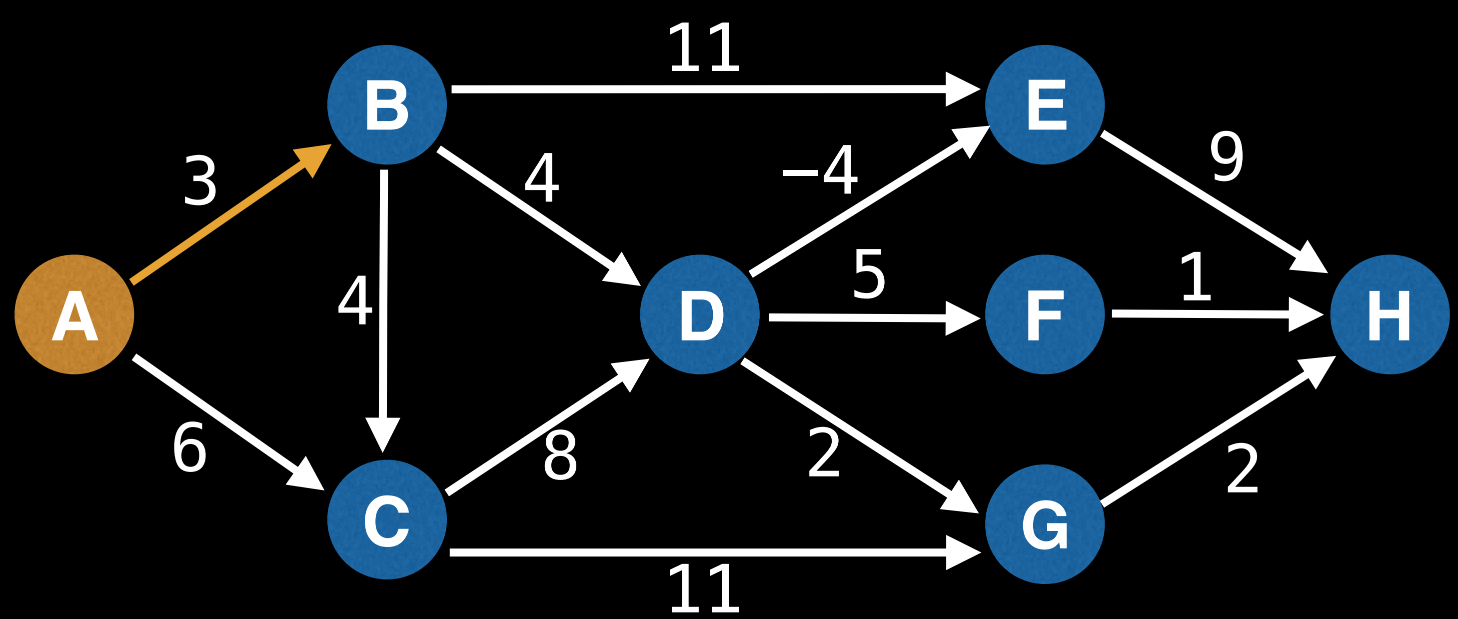


Arbitrary topological order: A, B, C, D, G, E, F, H

0	∞	∞	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

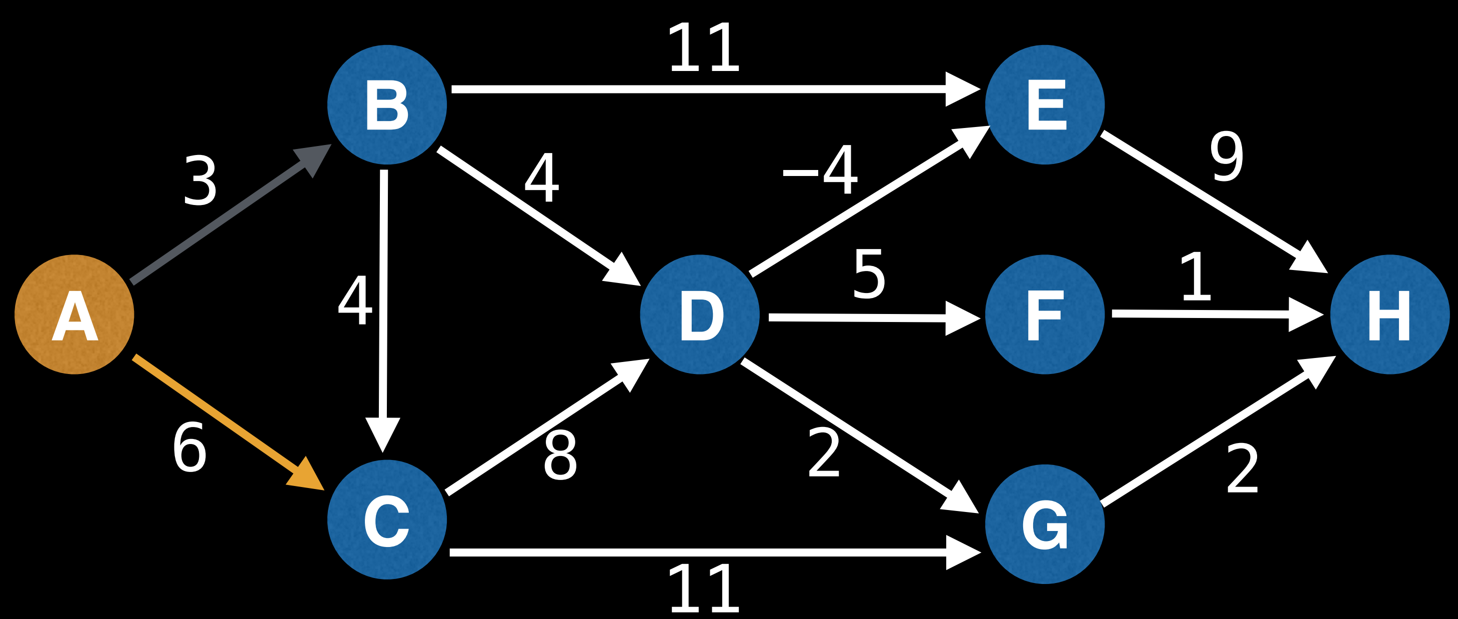


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	∞	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

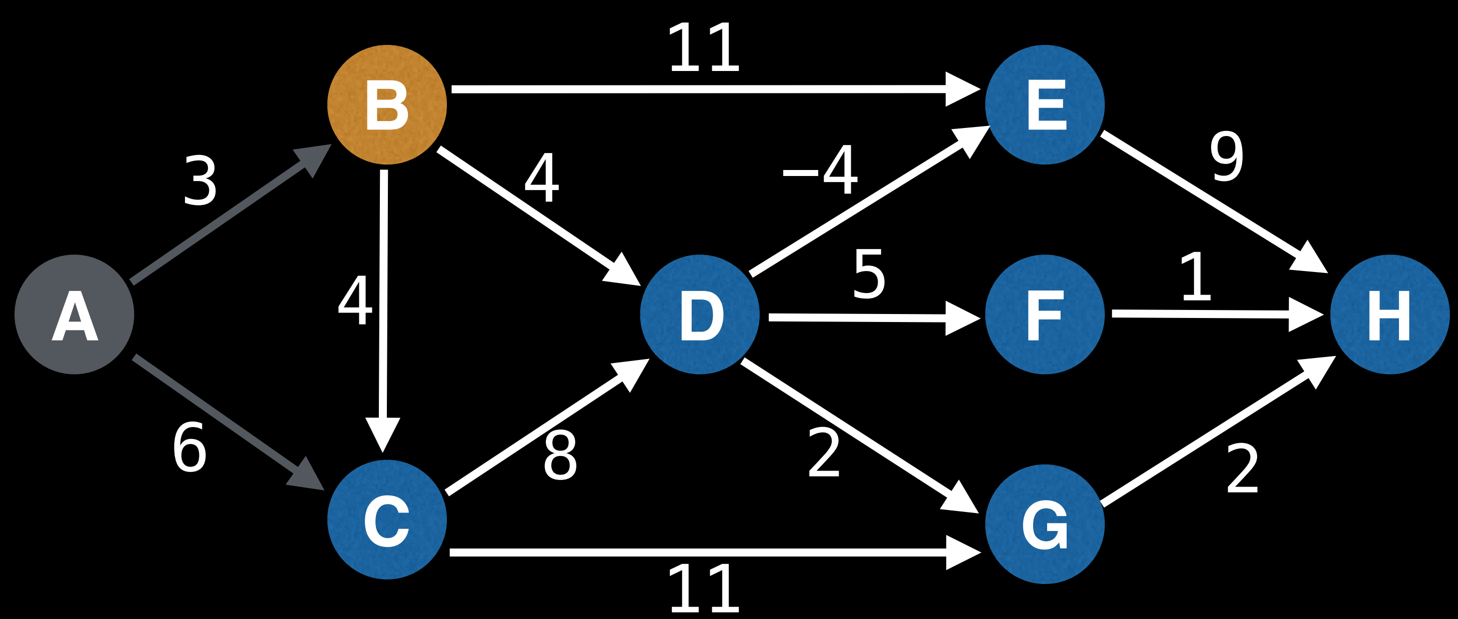


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

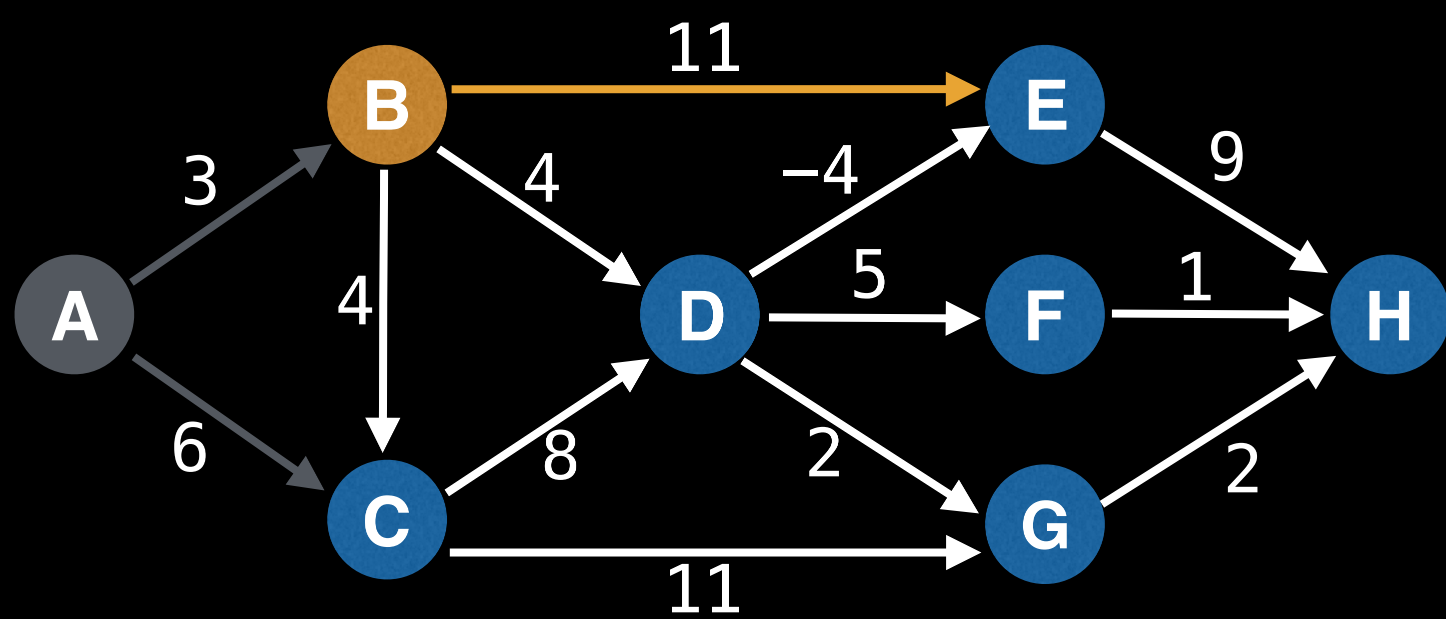


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	∞	∞	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

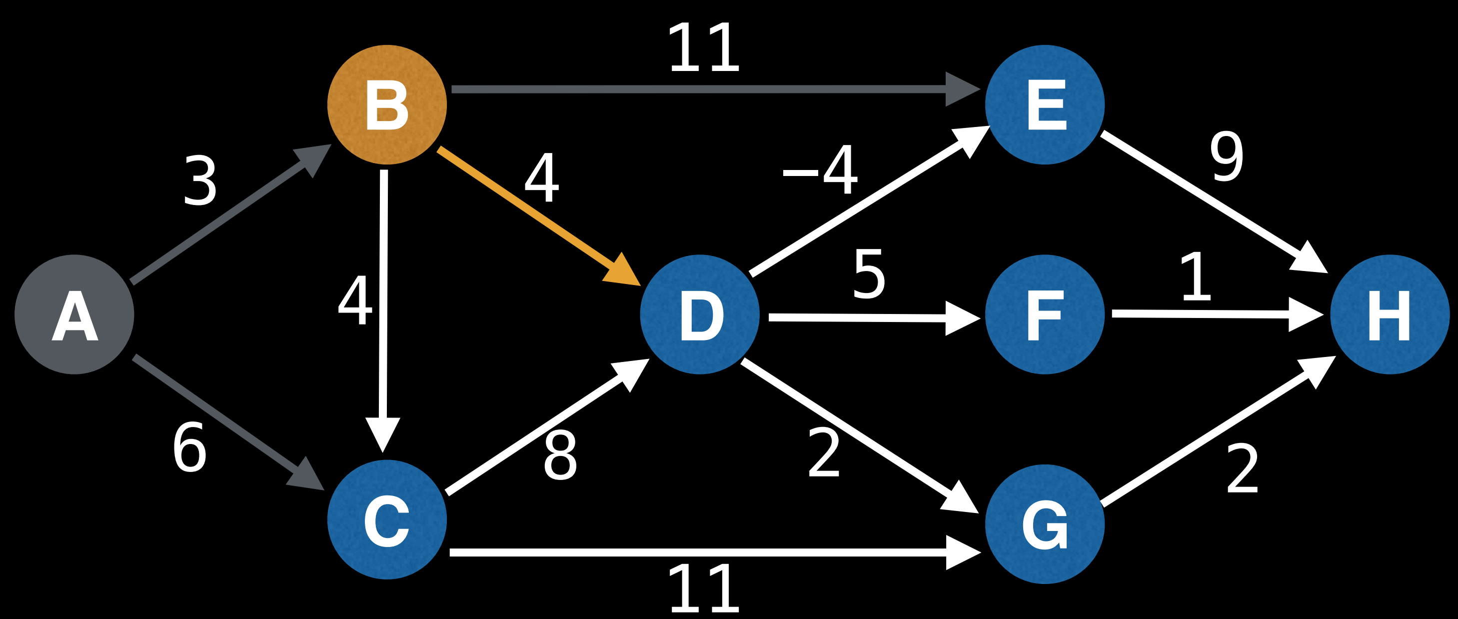


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	∞	14	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

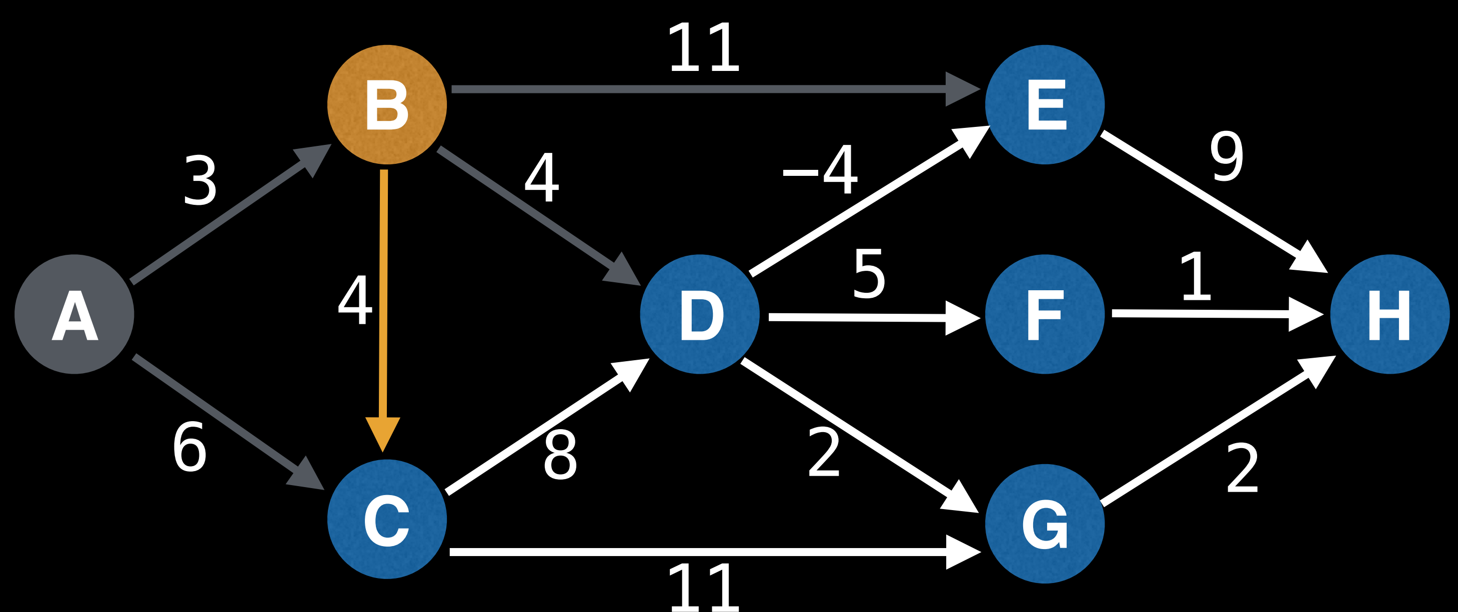


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

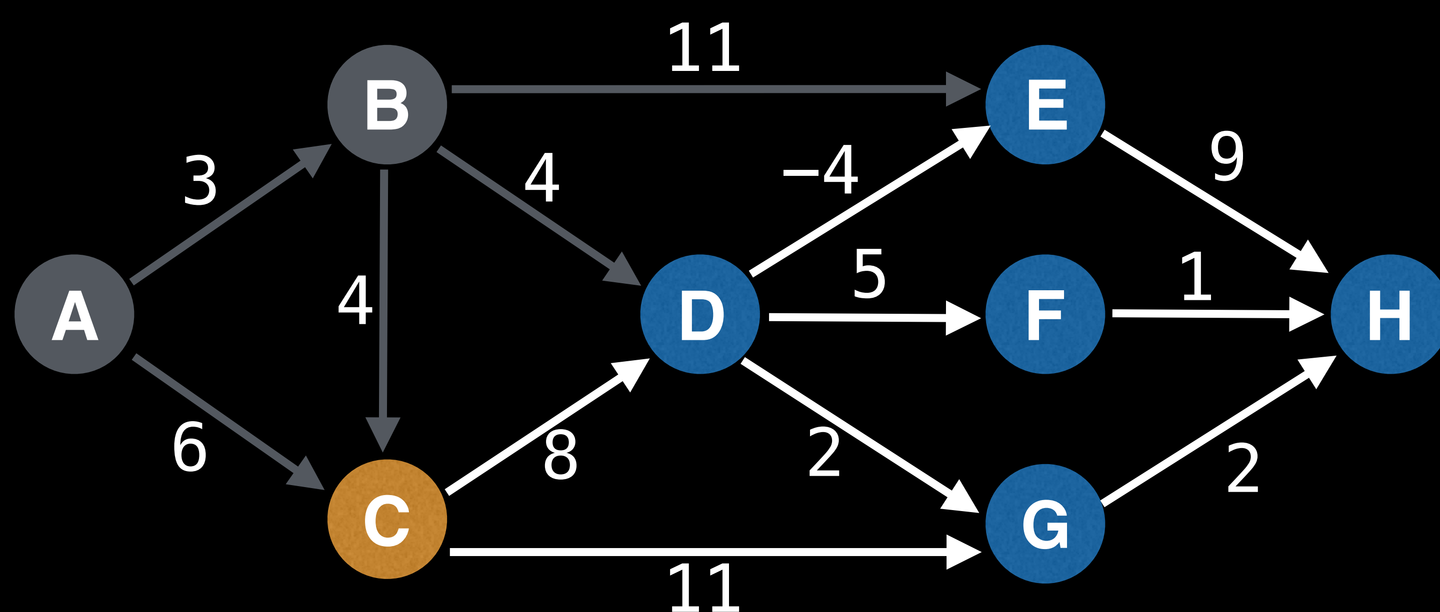


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

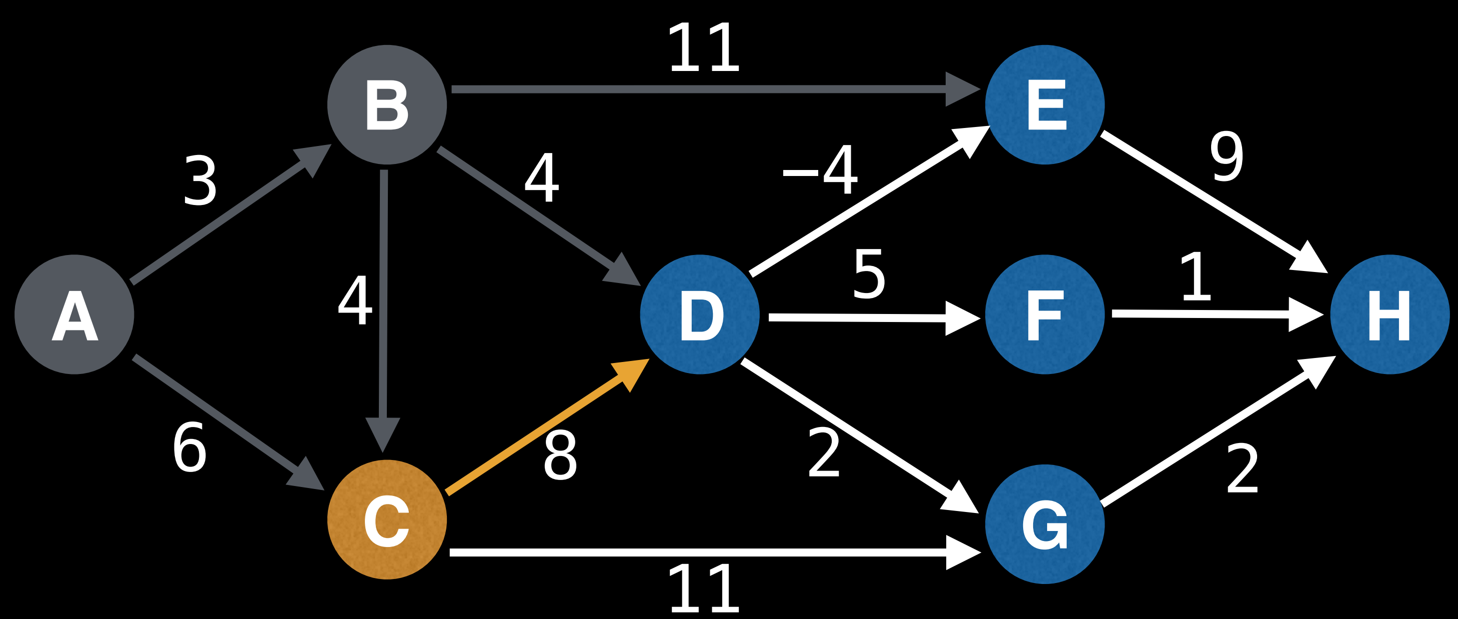


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

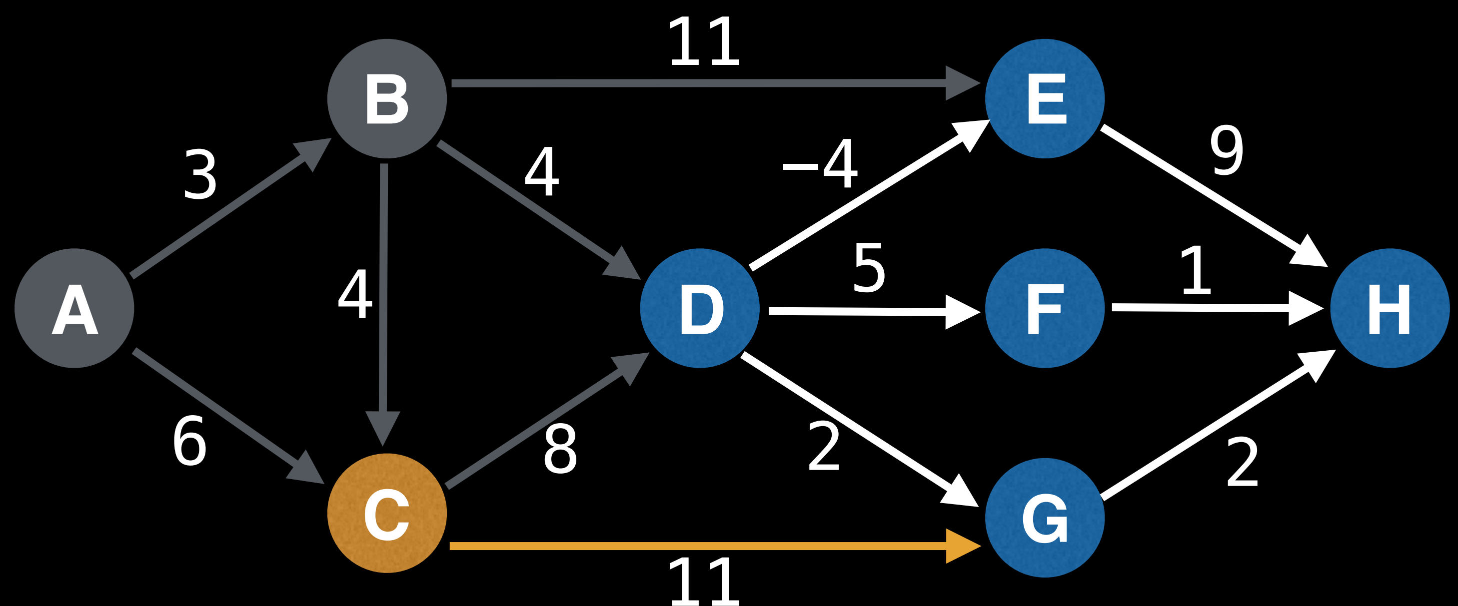


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	∞	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

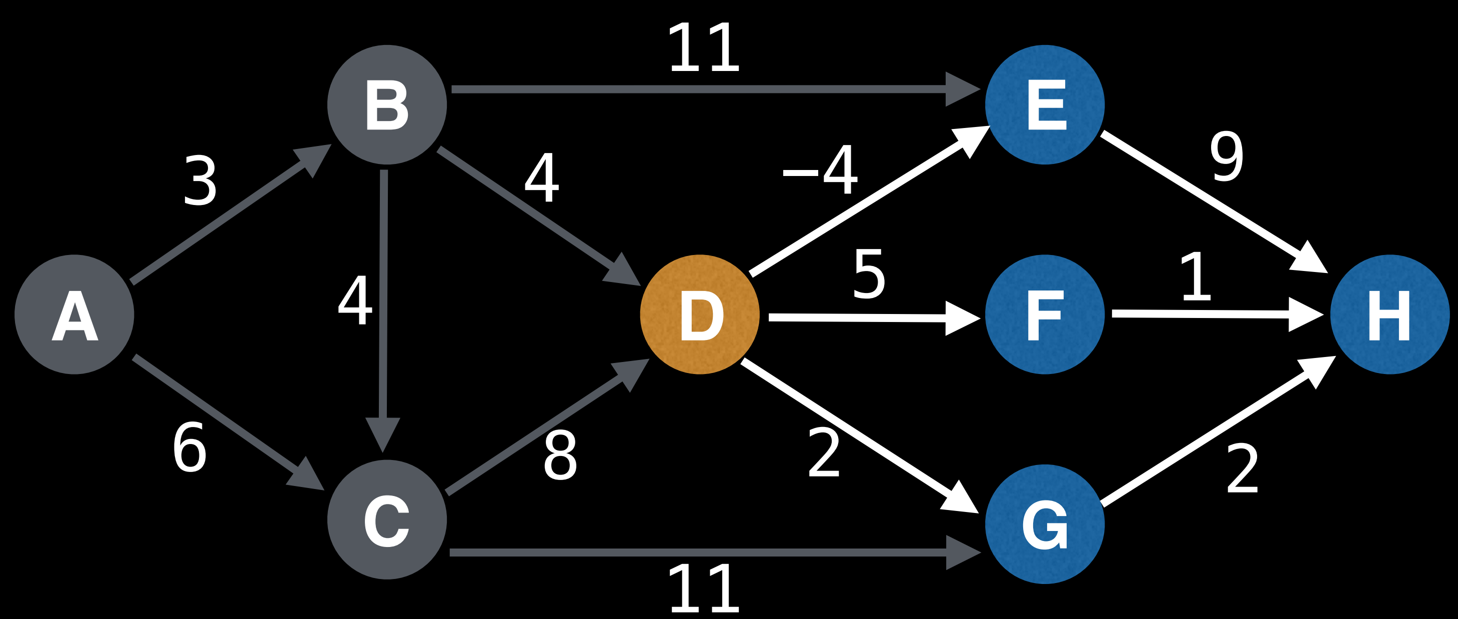


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	17	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

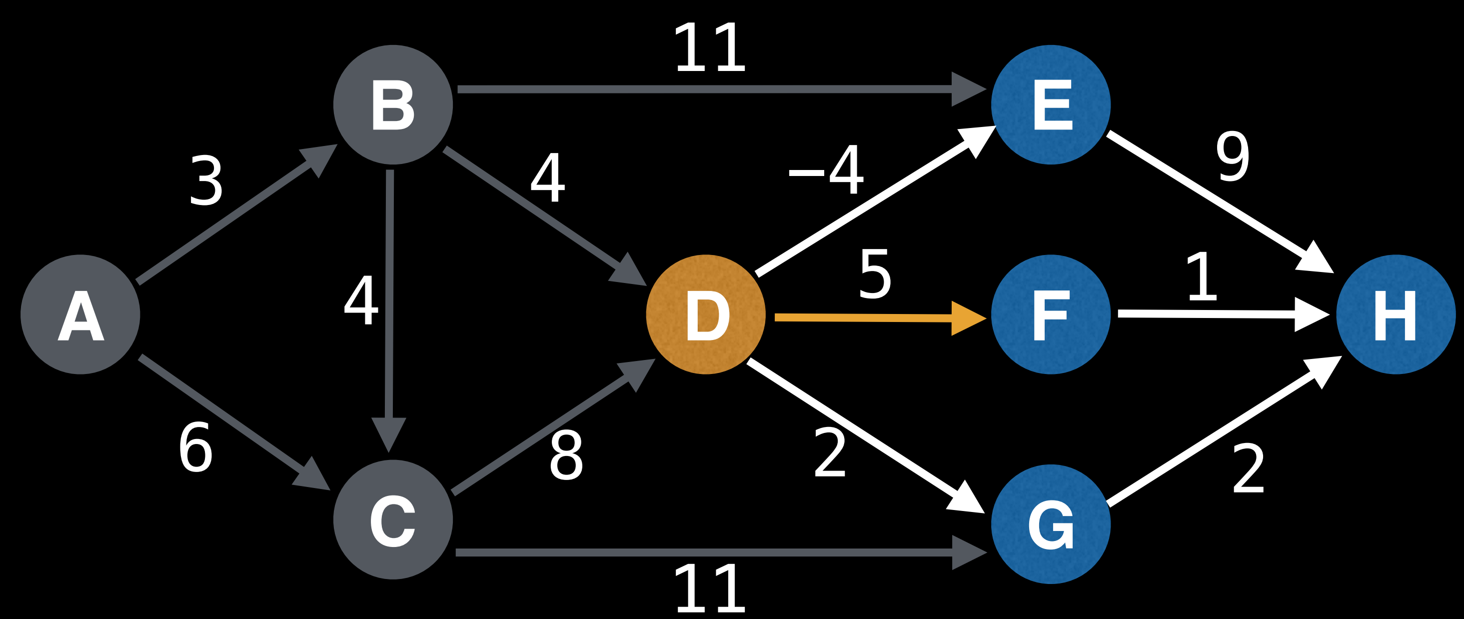


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	∞	17	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

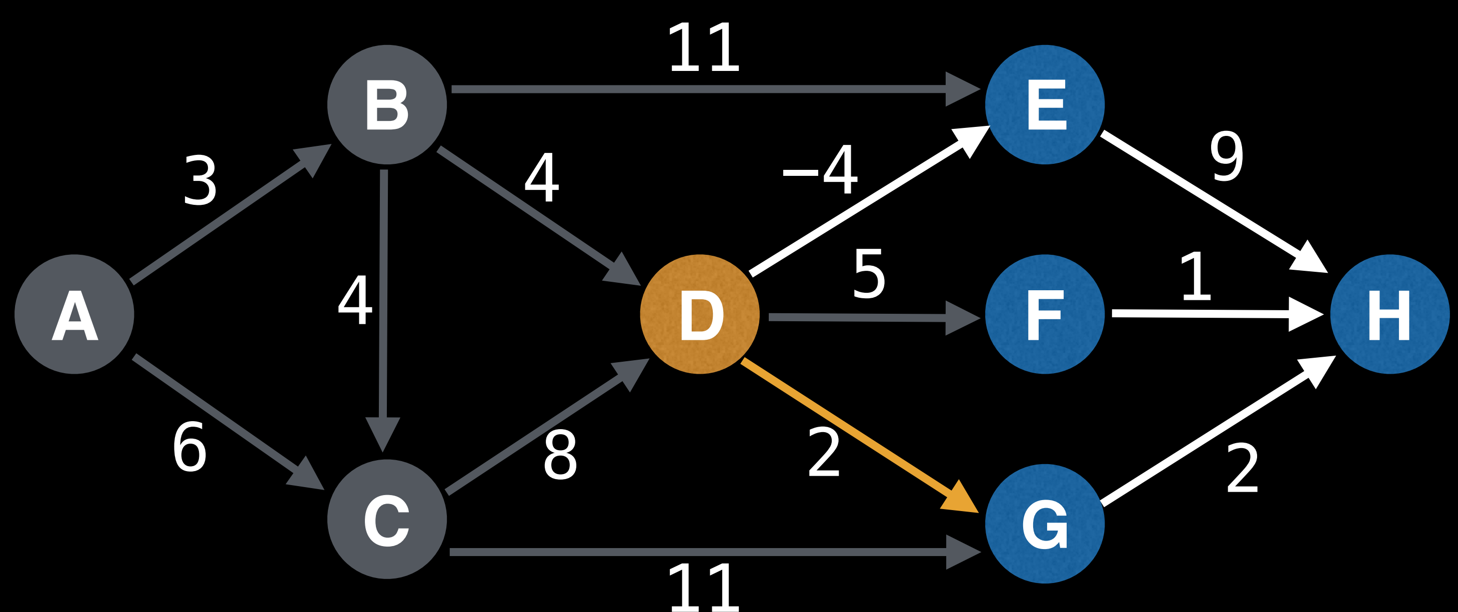


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	12	17	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

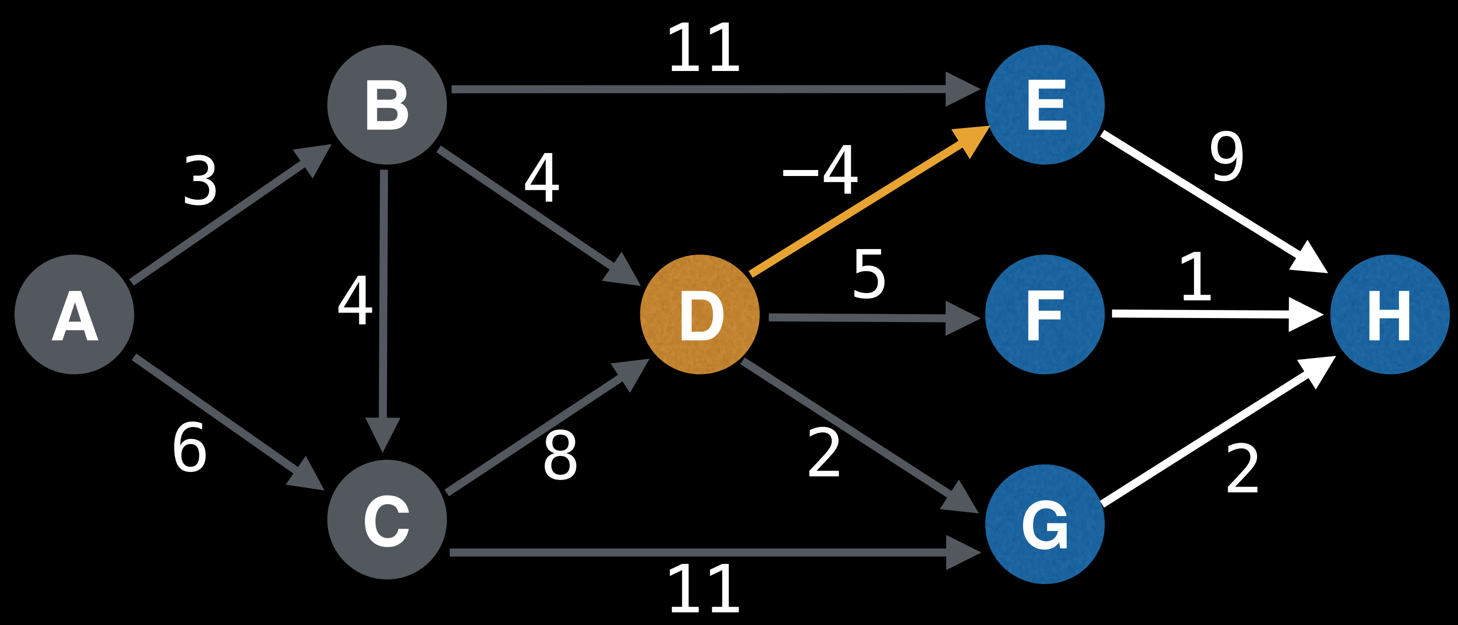


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	14	12	9	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

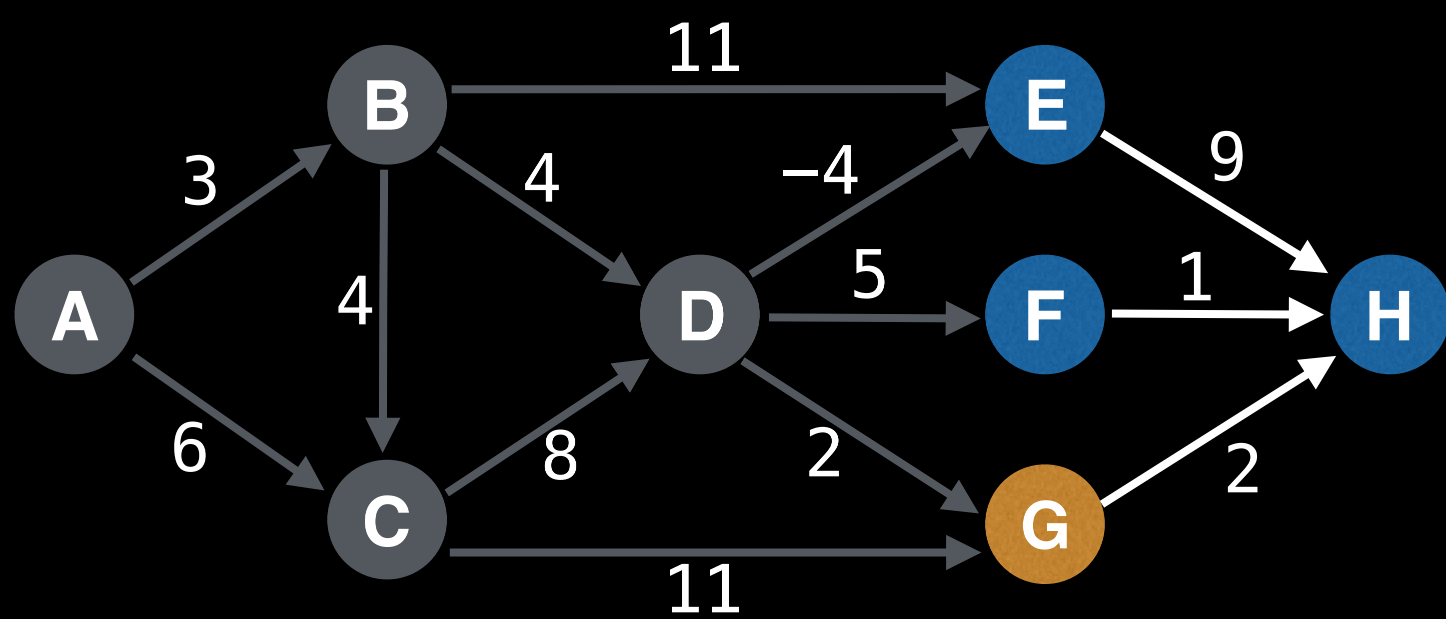


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

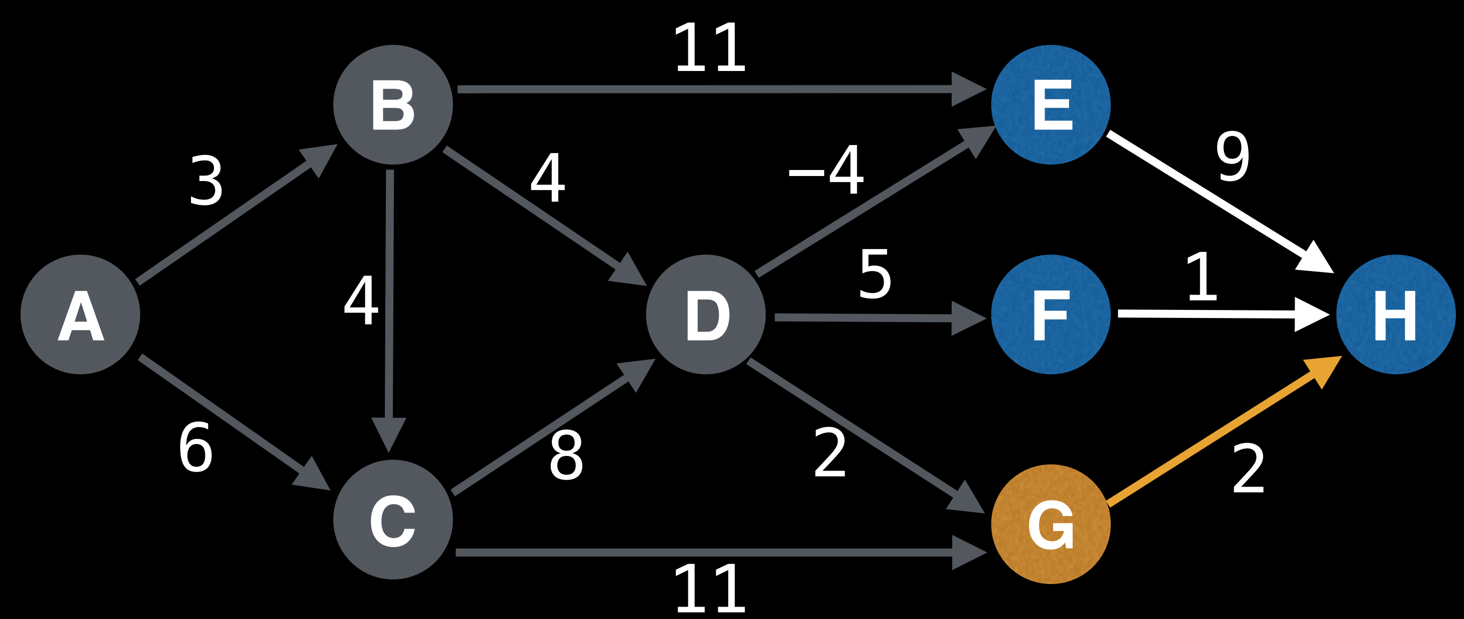


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	∞
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

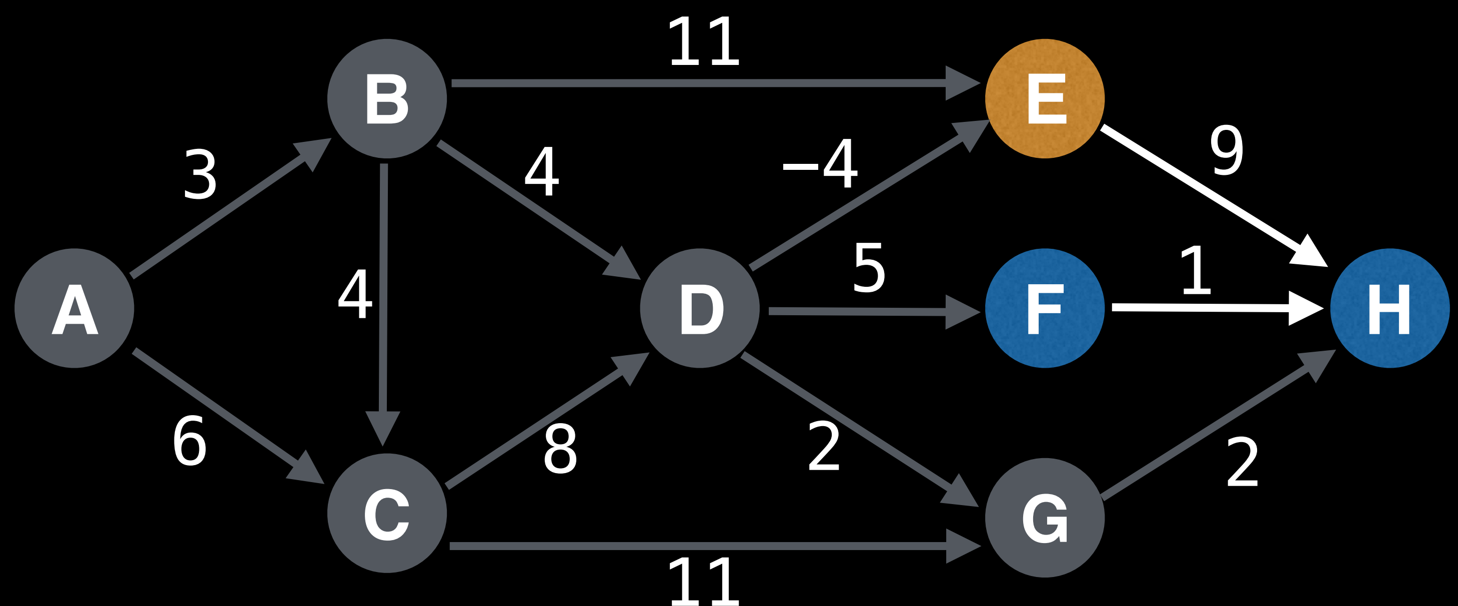


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

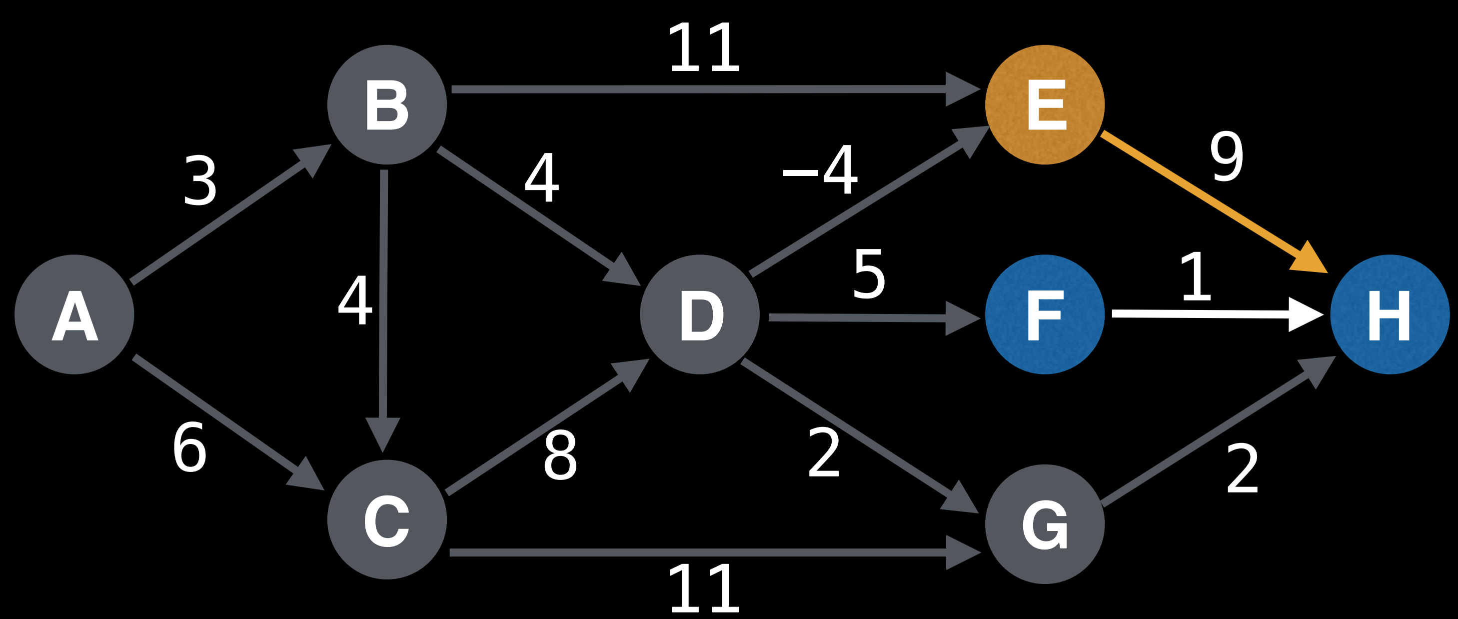


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

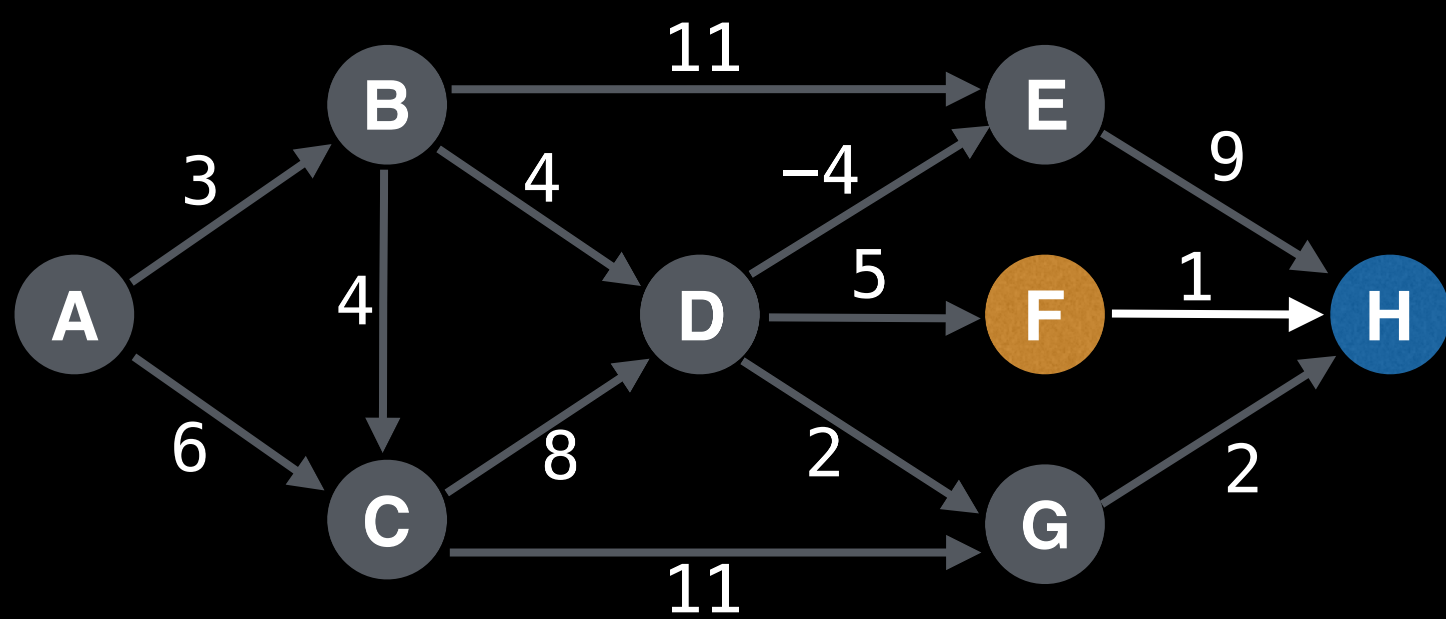


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

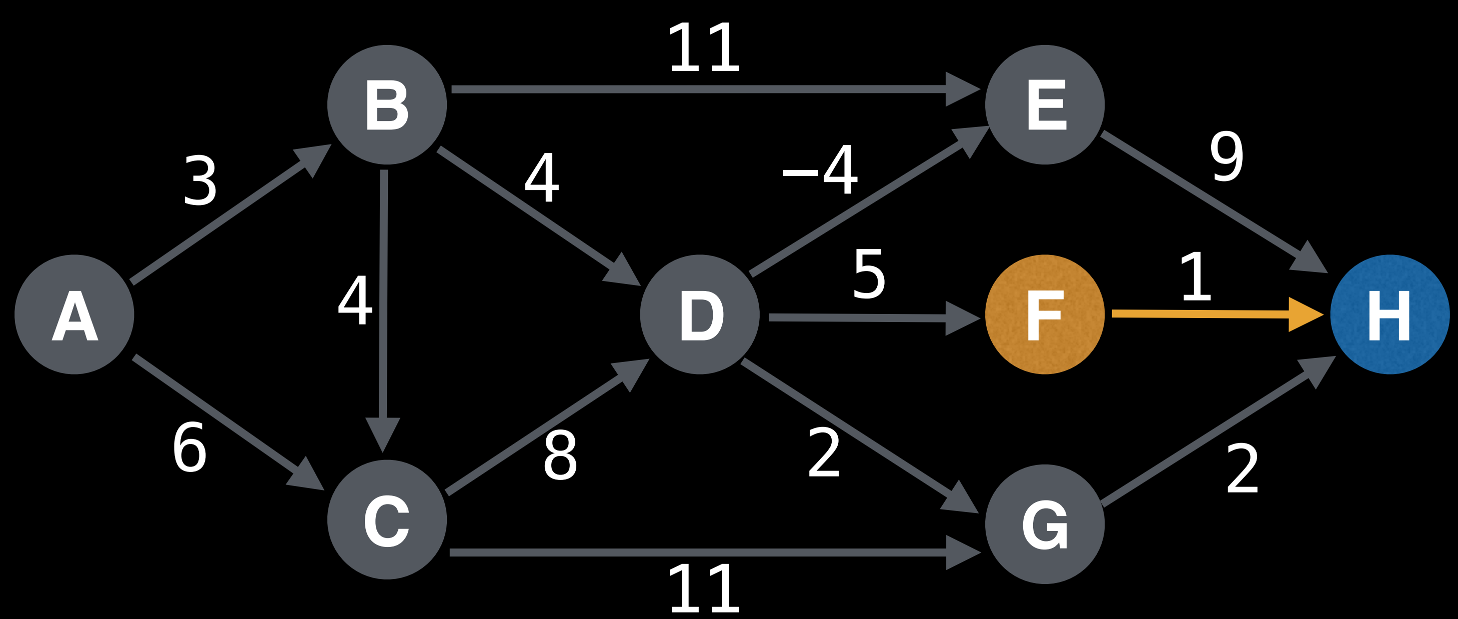


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

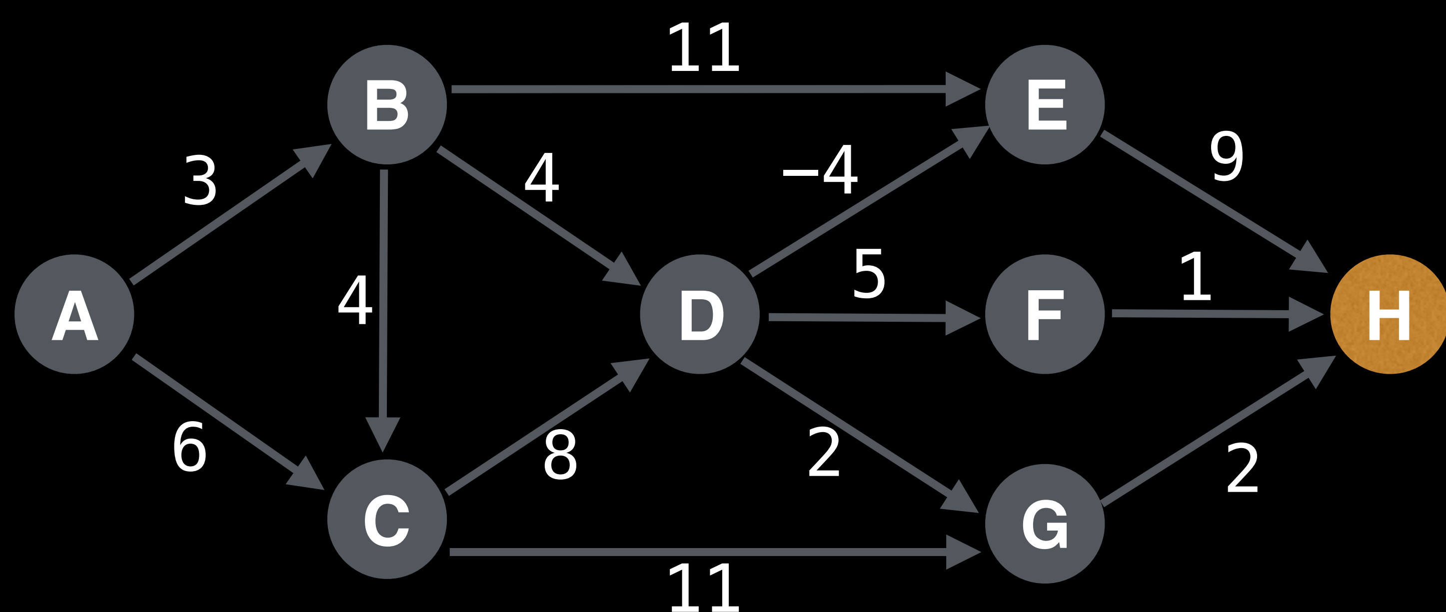


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

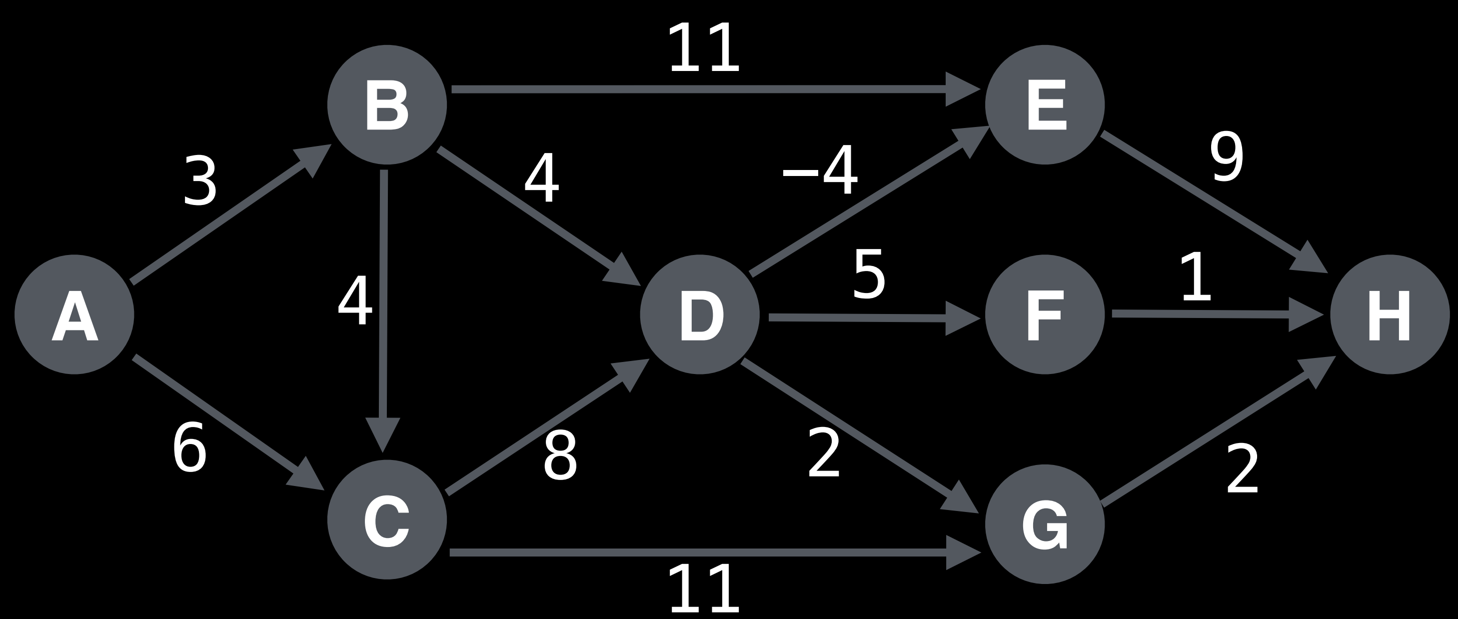


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.

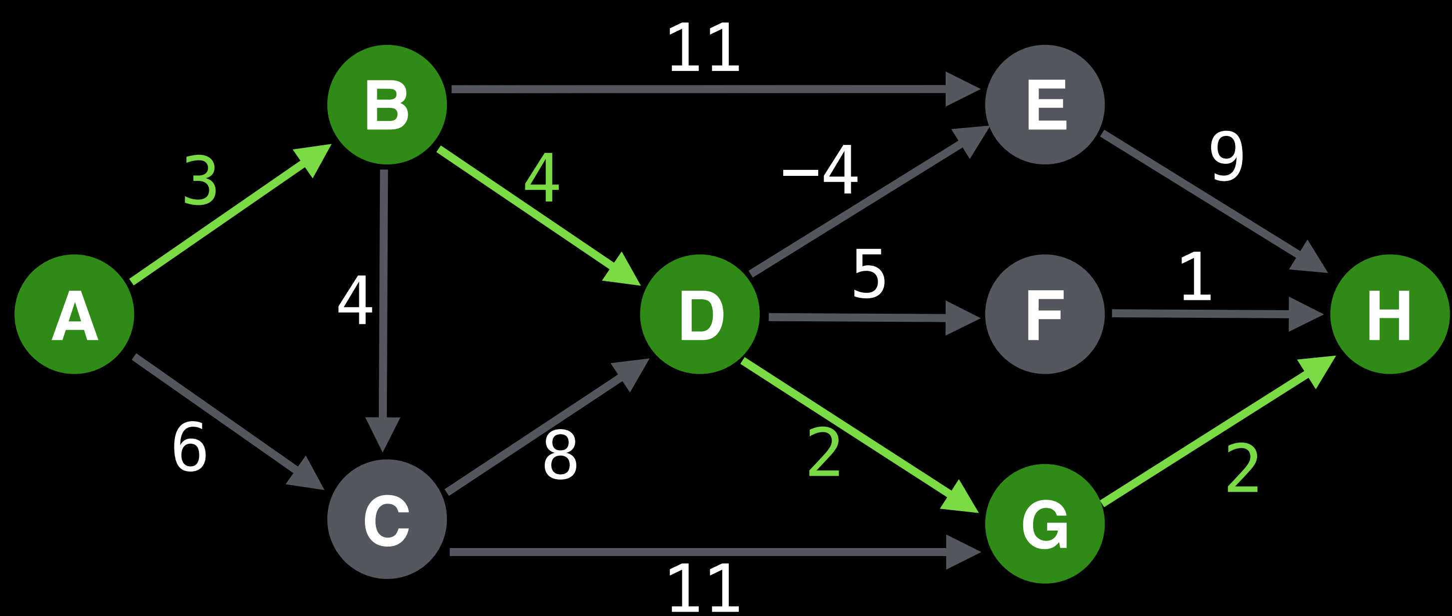


Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

SSSP on DAG

The **Single Source Shortest Path (SSSP)** problem can be solved efficiently on a DAG in **O(V+E)** time. This is due to the fact that the nodes can be ordered in a **topological ordering** via topsort and processed sequentially.



Arbitrary topological order: A, B, C, D, G, E, F, H

0	3	6	7	3	12	9	11
A	B	C	D	E	F	G	H

Longest path on DAG

What about the longest path? On a general graph this problem is **NP-Hard**, but on a DAG this problem is solvable in **$O(V+E)$**!

Longest path on DAG

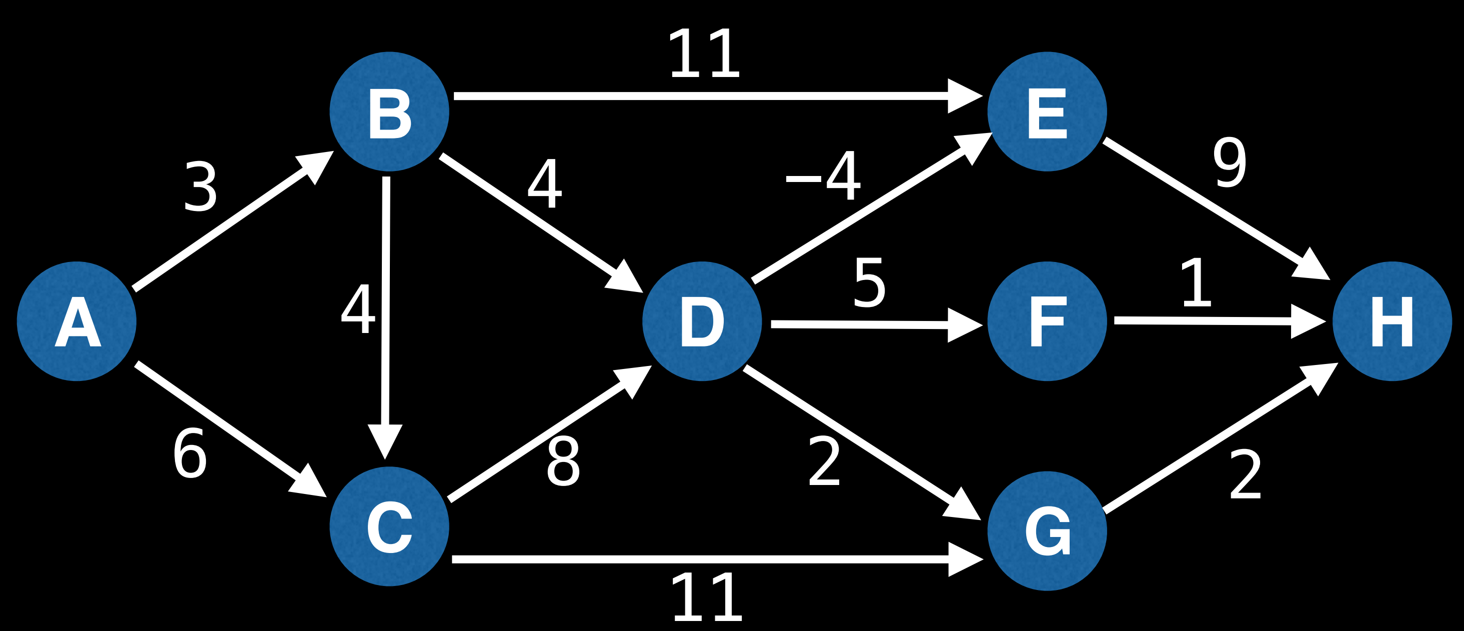
What about the longest path? On a general graph this problem is **NP-Hard**, but on a DAG this problem is solvable in **O(V+E)**!

The trick is to multiply all edge values by −1 then find the shortest path and then multiply the edge values by −1 again!

Longest path on DAG

What about the longest path? On a general graph this problem is **NP-Hard**, but on a DAG this problem is solvable in **O(V+E)**!

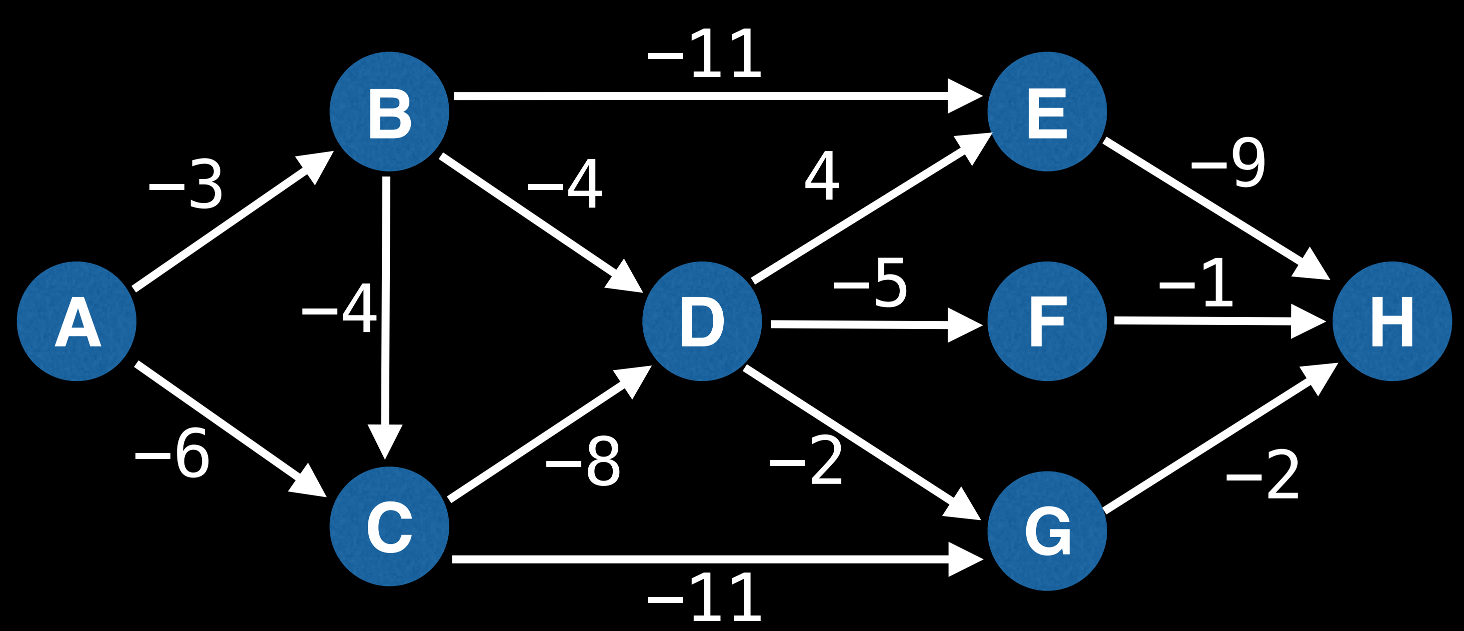
The trick is to multiply all edge values by −1 then find the shortest path and then multiply the edge values by −1 again!



Longest path on DAG

What about the longest path? On a general graph this problem is **NP-Hard**, but on a DAG this problem is solvable in **O(V+E)**!

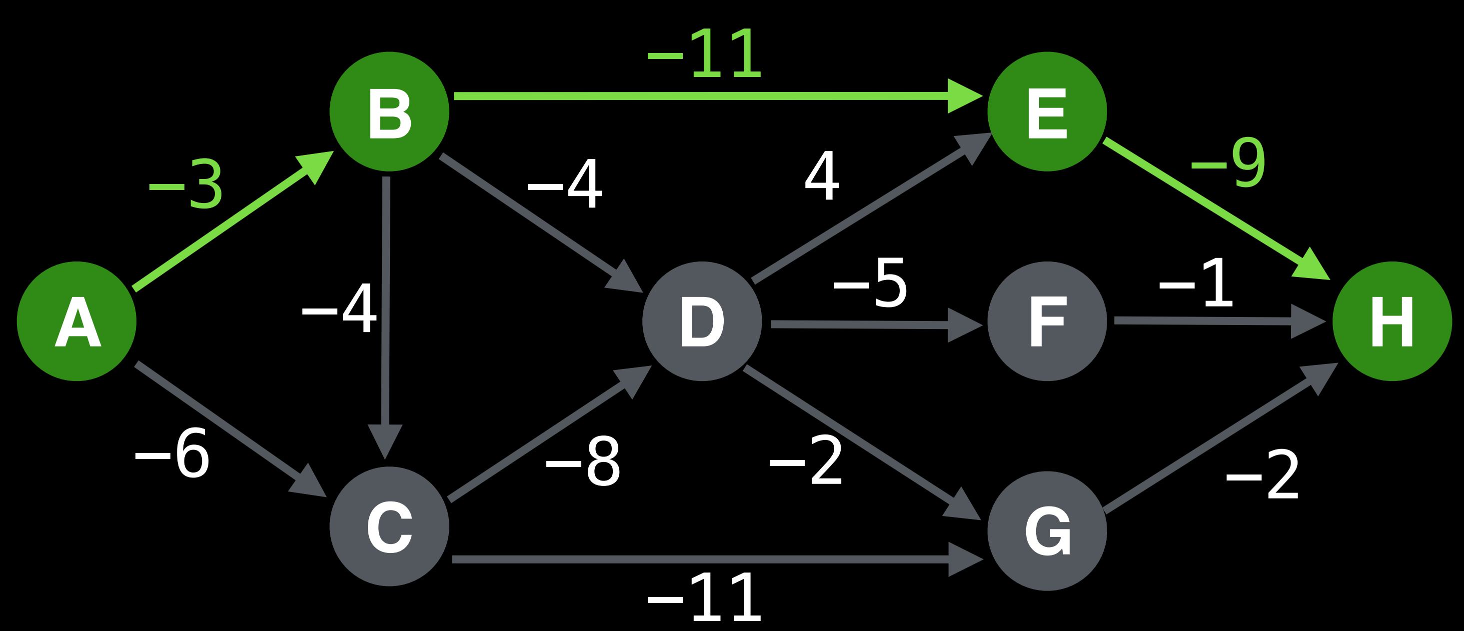
The trick is to multiply all edge values by −1 then find the shortest path and then multiply the edge values by −1 again!



Longest path on DAG

What about the longest path? On a general graph this problem is **NP-Hard**, but on a DAG this problem is solvable in **O(V+E)**!

The trick is to multiply all edge values by −1 then find the shortest path and then multiply the edge values by −1 again!



$$(-3 + -11 + -9) * -1 = 23$$

Source Code Link

Implementation source code can be found at the following link:

github.com/williamfiset/algorithms

Link in the description: