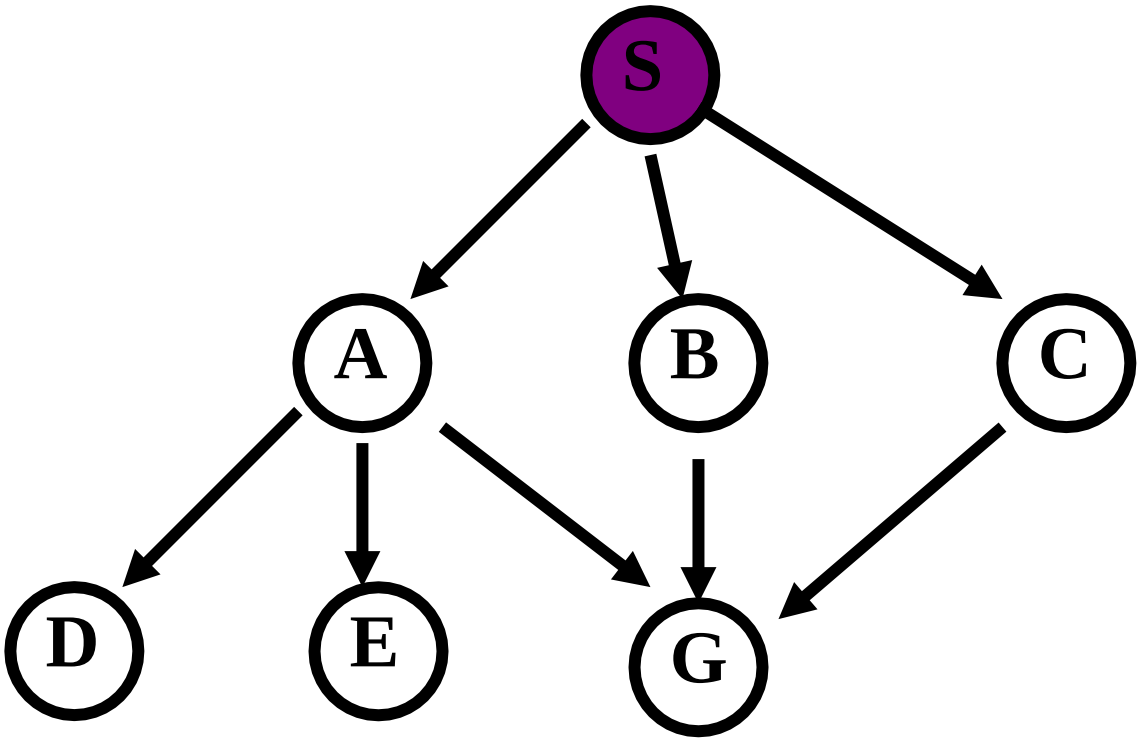
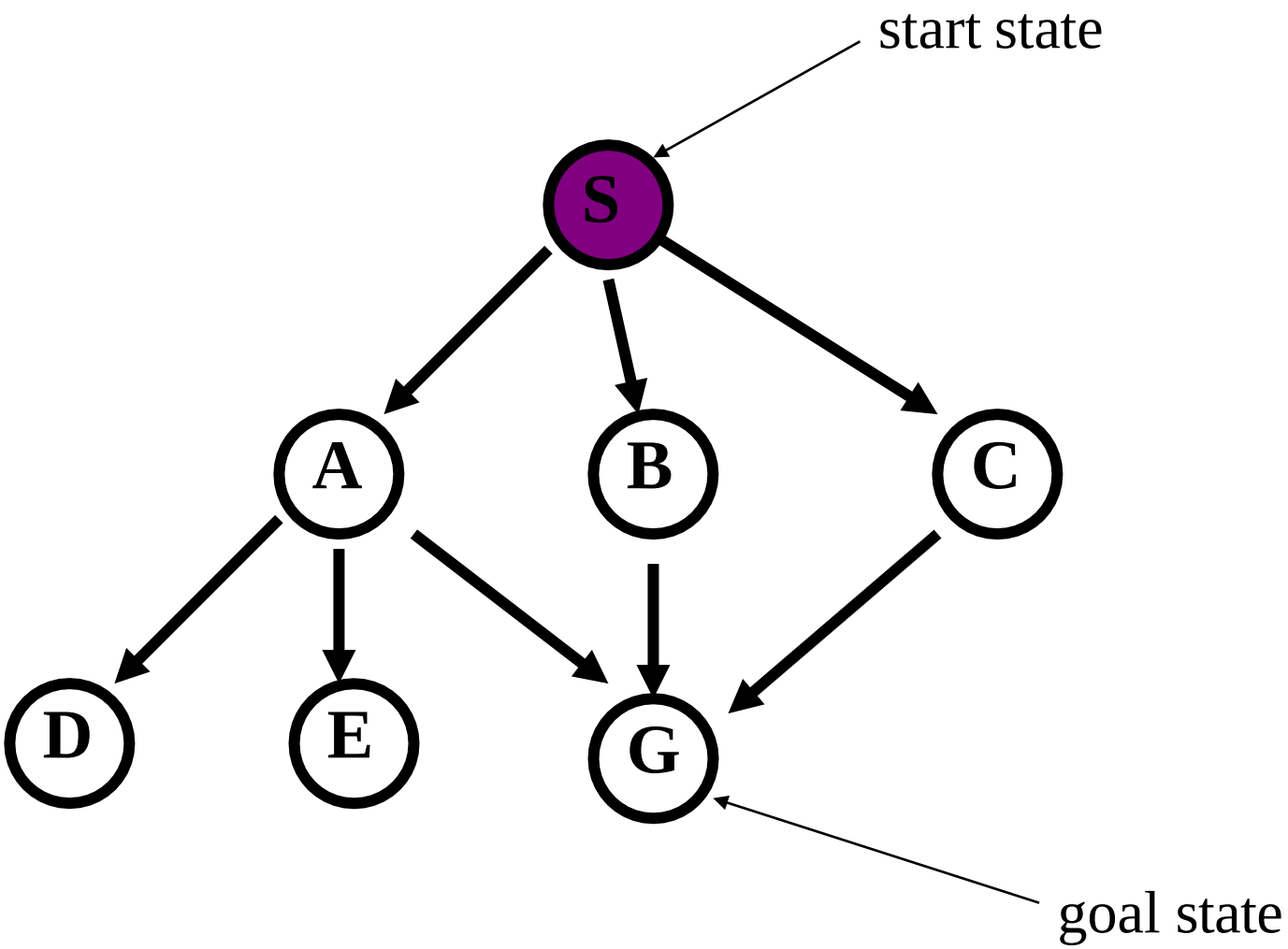
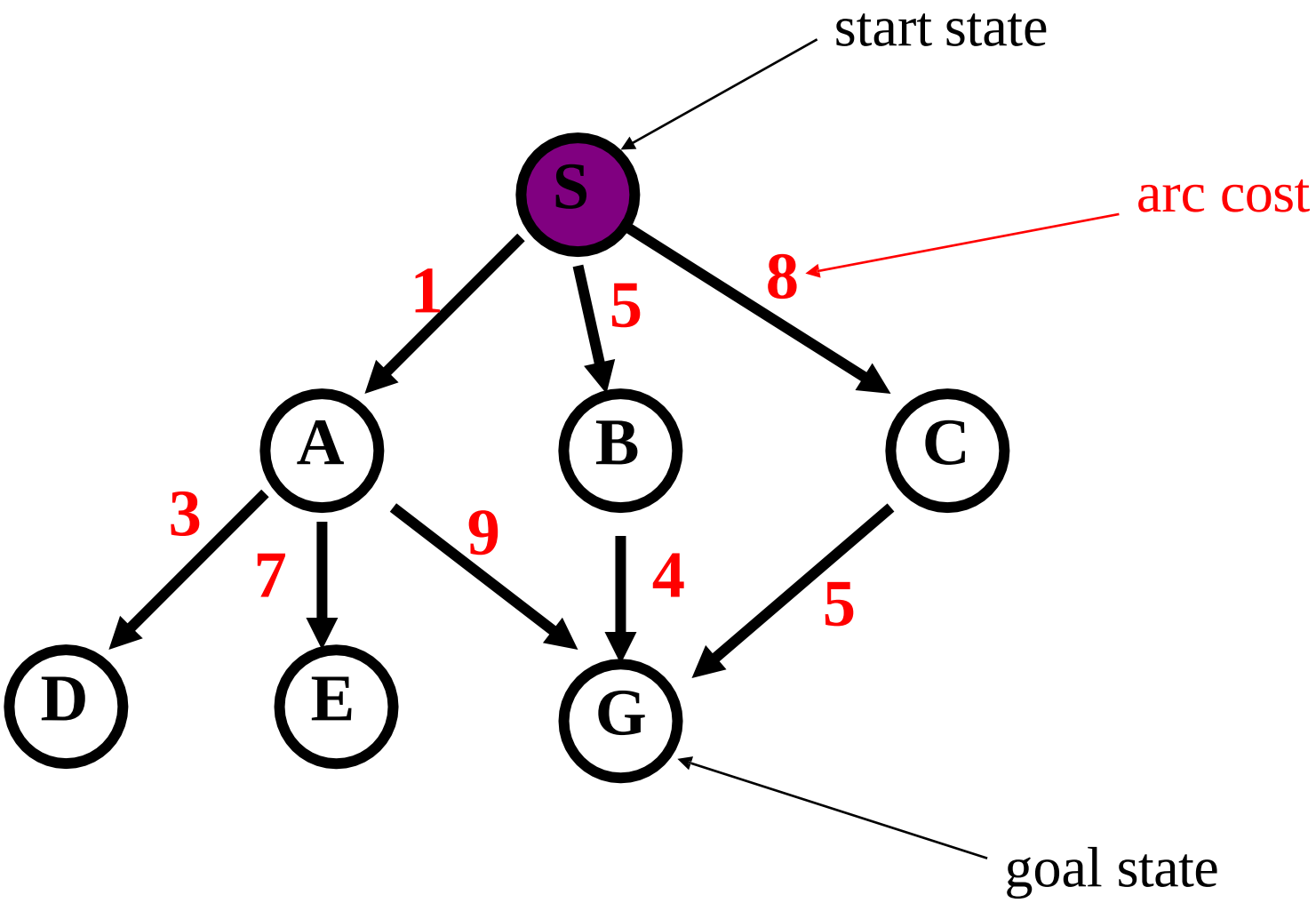
# Example search space



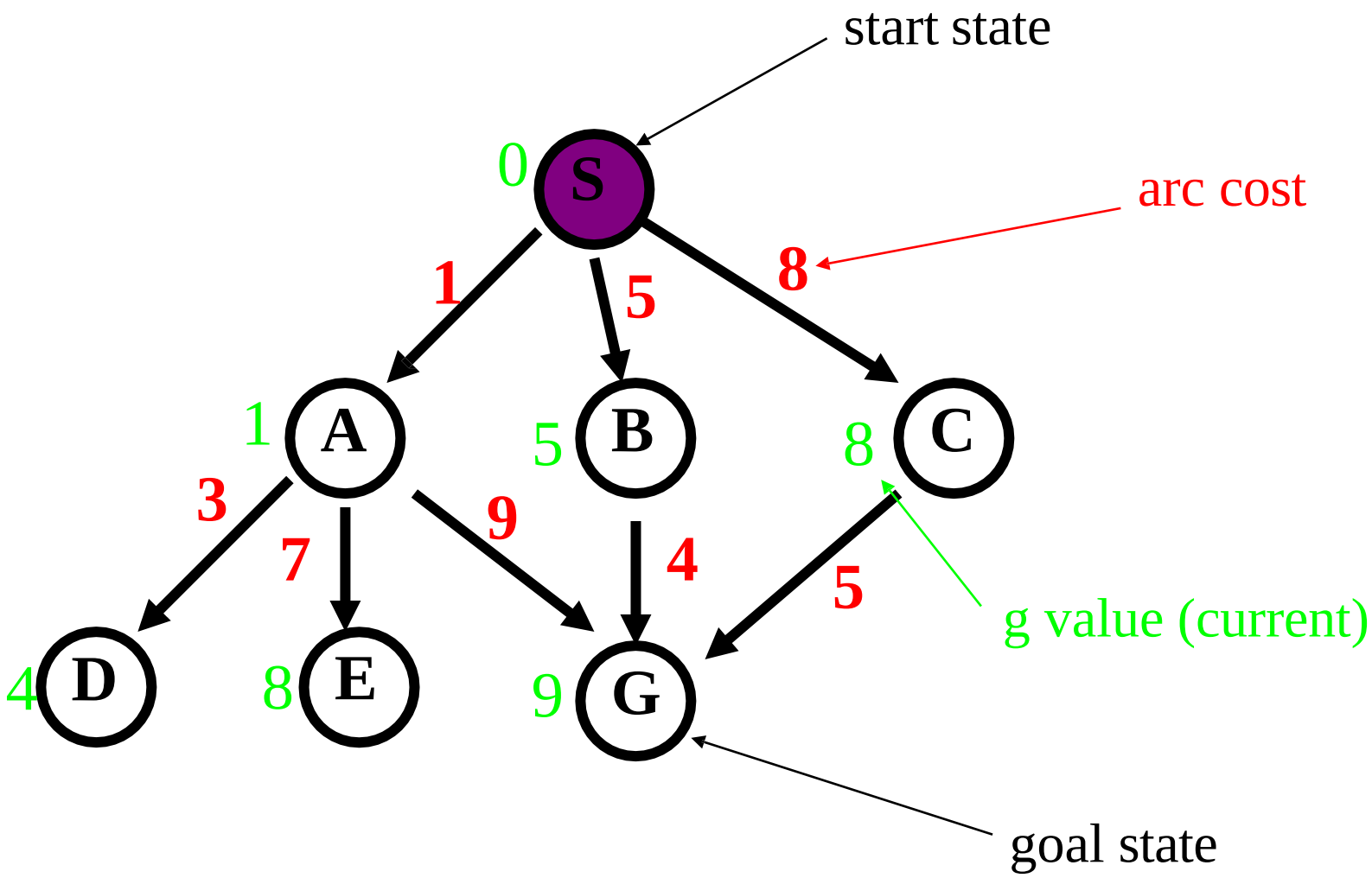
# Example search space



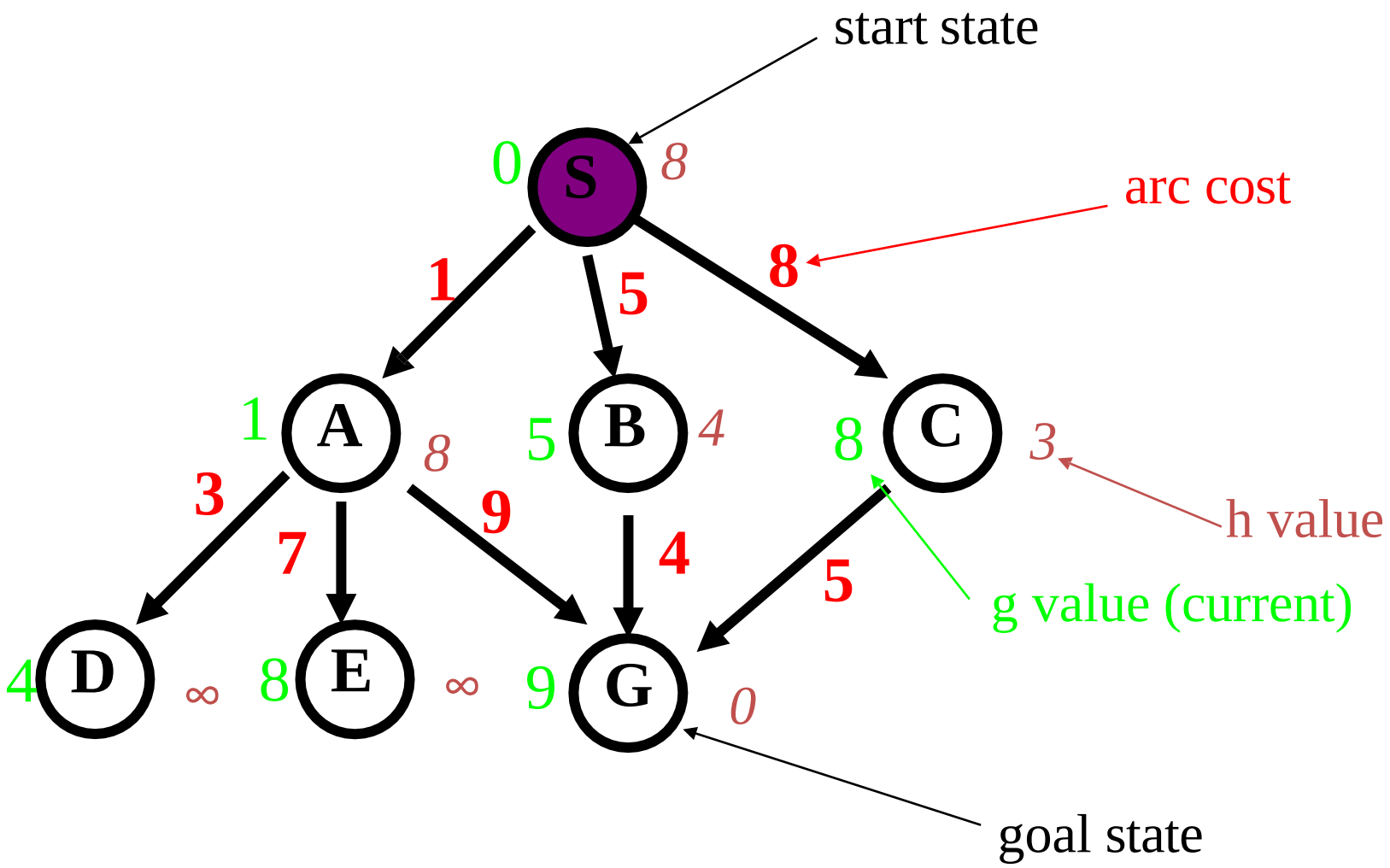
# Example search space



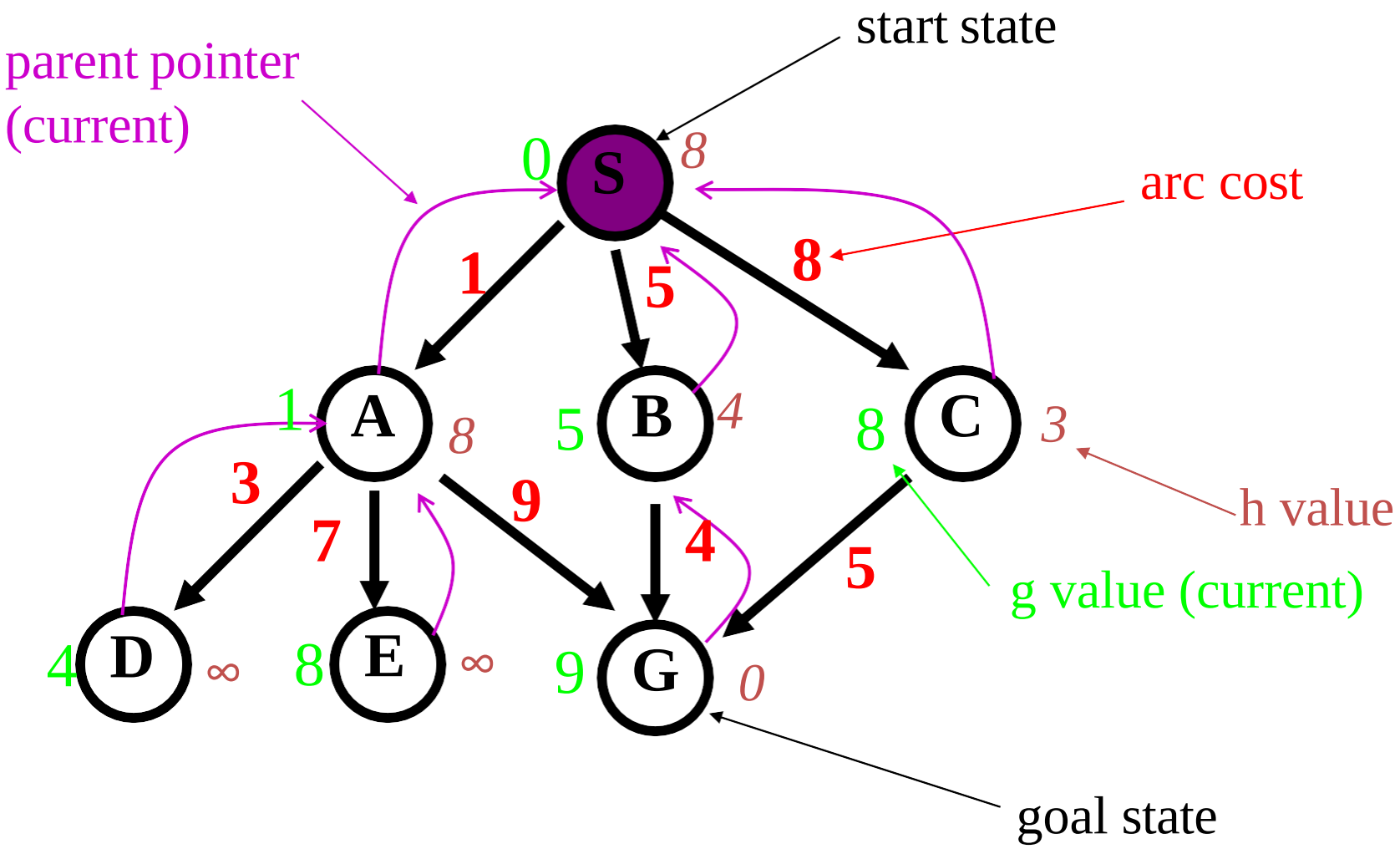
# Example search space



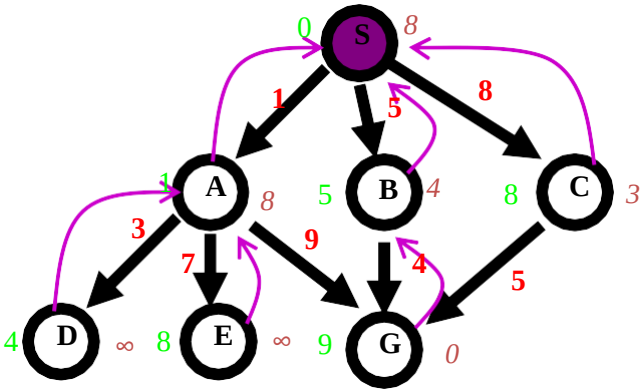
# Example search space



# Example search space



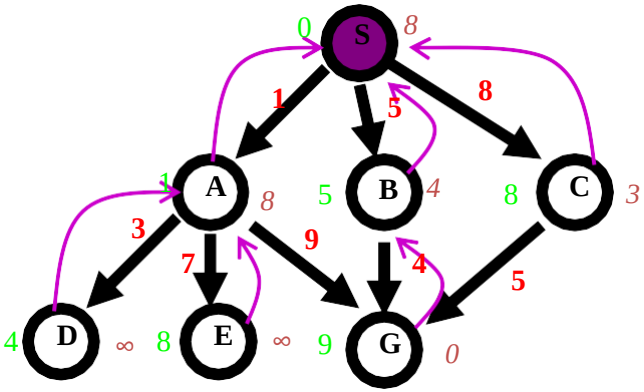
# Example




| n | g(n) | h(n) | f(n) | h*(n) |
|---|---|---|---|---|
| S | 0 | 8 | 8 | 9 |

- h*(n) is (hypothetical) perfect heuristic (an oracle)
- Since h(n) <= h*(n) for all n, h is admissible (optimal)
- Optimal path = *SBG* with cost 9

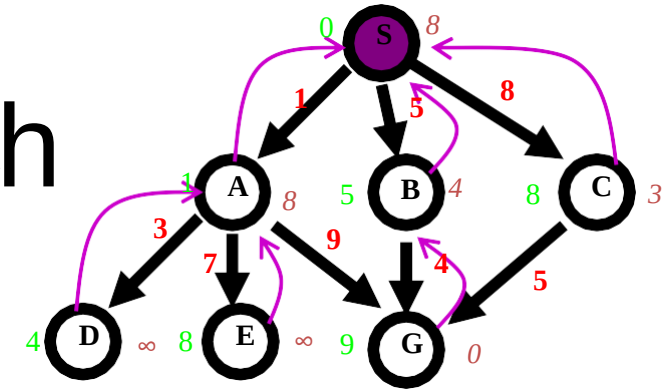
# Example




| n | g(n) | h(n) | f(n) | h*(n) |
|---|---|---|---|---|
| S | 0 | 8 | 8 | 9 |
| A | 1 | 8 | 9 | 9 |
| B | 5 | 4 | 9 | 4 |
| C | 8 | 3 | 11 | 5 |
| D | 4 | inf | inf | inf |
| E | 8 | inf | inf | inf |
| G | 9 | 0 | 9 | 0 |

- h*(n) is (hypothetical) perfect heuristic (an oracle)
- Since h(n) <= h*(n) for all n, h is admissible (optimal)
- Optimal path = *SBG* with cost 9

# Greedy search



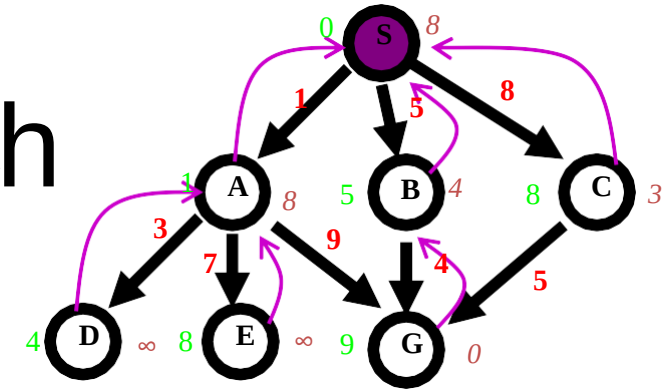

f(n) = h(n)

**node expanded** **nodes list**

{ S(8) }

what's next???

# Greedy search

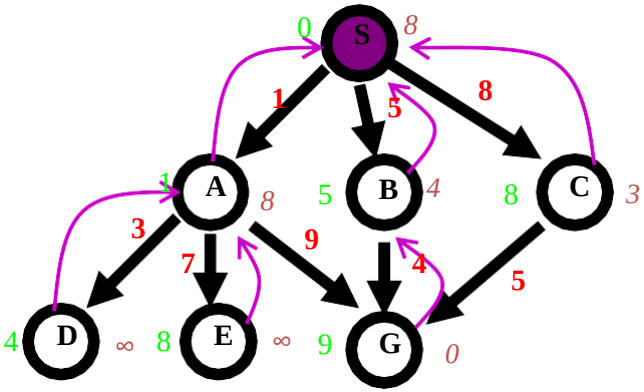



f(n) = h(n)

| **node expanded** | **nodes list** |
|---|---|
| | { S(8) } |
| S | { C(3) B(4) A(8) } |
| C | { G(0) B(4) A(8) } |
| G | { B(4) A(8) } |

- Solution path found is SCG, 3 nodes expanded.
- See how fast the search is!! But it is NOT optimal.

# A* search



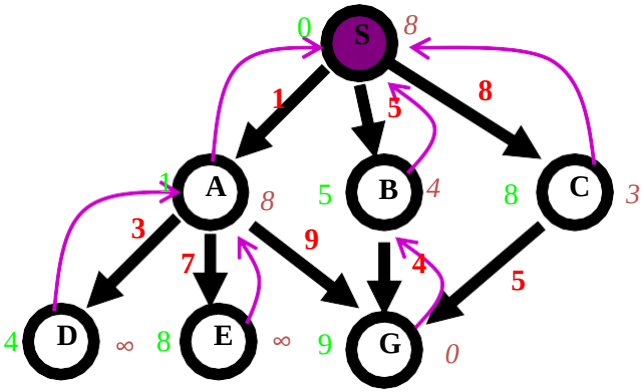

**f(n) = g(n) + h(n)**

node exp. nodes list

{ S(8) }

What's next?

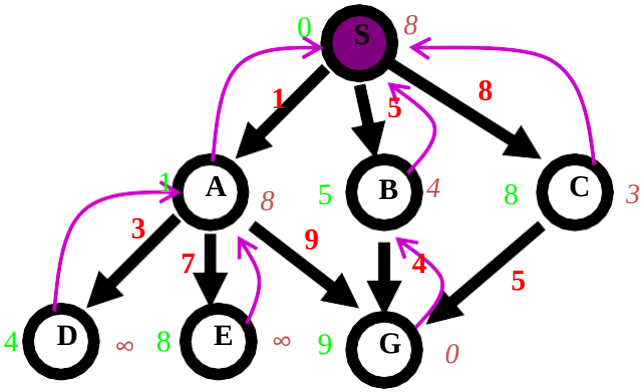
# A* search




**f(n) = g(n) + h(n)**

```
node exp.      nodes list
               { S(8) }
 S             { A(9) B(9) C(11) }
                     What's next?
```

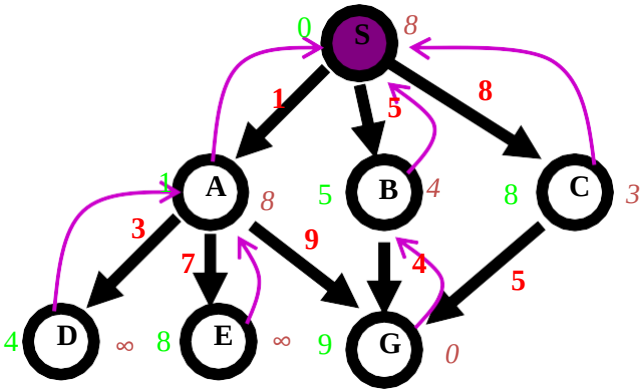
# A* search




**f(n) = g(n) + h(n)**

```
node exp.      nodes list
               { S(8) }
 S             { A(9) B(9) C(11) }
 A             { B(9) G(10) C(11) D(inf) E(inf) }
                    What's next?
```

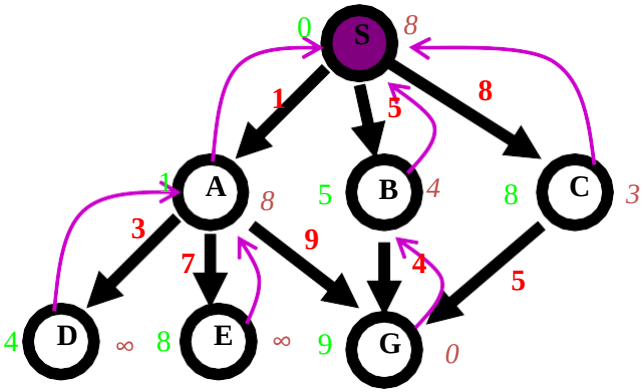
# A* search




**f(n) = g(n) + h(n)**

```
node exp.      nodes list
               { S(8) }
 S             { A(9) B(9) C(11) }
 A             { B(9) G(10) C(11) D(inf) E(inf) }
 B             { G(9) G(10) C(11) D(inf) E(inf) }
                    What's next?
```

# A* search




**f(n) = g(n) + h(n)**

```
node exp.      nodes list
               { S(8) }
 S             { A(9) B(9) C(11) }
 A             { B(9) G(10) C(11) D(inf) E(inf) }
 B             { G(9) G(10) C(11) D(inf) E(inf) }
 G             { C(11) D(inf) E(inf) }
```

- Solution path found is S B G, 4 nodes expanded..
- Still pretty fast. And optimal, too.