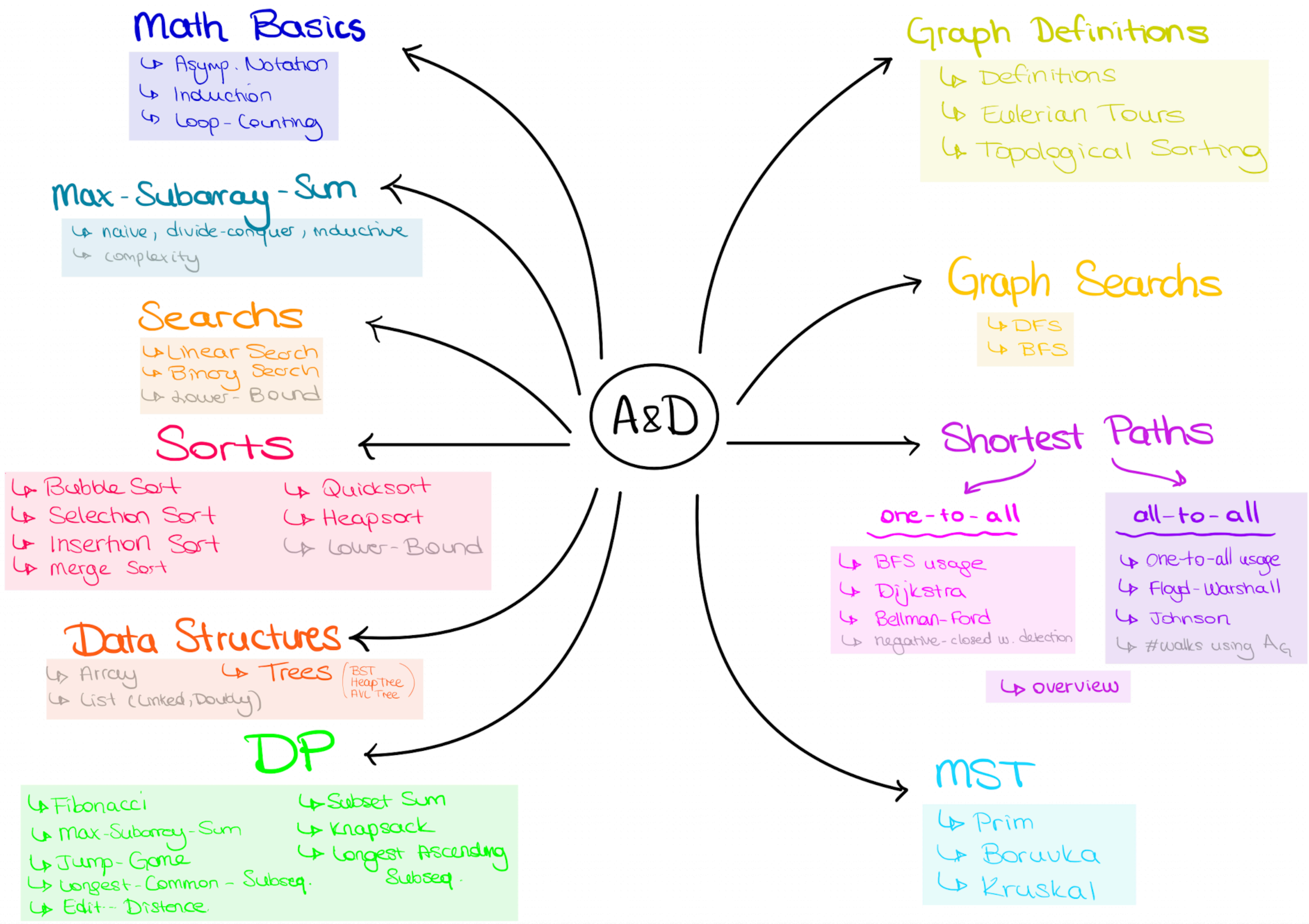
A&D

Exercise Session 10

Nil Ozer

A&D Overview



A&D

Math Basics
- ↳ Asymp. Notation
- ↳ Induction
- ↳ Loop-Counting

Max-Subarray-Sum
- ↳ naive, divide-conquer, inductive
- ↳ complexity

Searchs
- ↳ Linear Search
- ↳ Binary Search
- ↳ Lower-Bound

Sorts
- ↳ Bubble Sort
- ↳ Selection Sort
- ↳ Insertion Sort
- ↳ merge Sort
- ↳ Quicksort
- ↳ Heapsort
- ↳ Lower-Bound

Data Structures
- ↳ Array
- ↳ List (Linked, Doubly)
- ↳ Trees (BST, Heap Tree, AVL Tree)

DP
- ↳ Fibonacci
- ↳ Max-Subarray-Sum
- ↳ Jump-Game
- ↳ Longest-Common-Subseq.
- ↳ Edit-Distance
- ↳ Subset Sum
- ↳ Knapsack
- ↳ Longest Ascending Subseq.

Graph Definitions
- ↳ Definitions
- ↳ Eulerian Tours
- ↳ Topological Sorting

Graph Searchs
- ↳ DFS
- ↳ BFS

Shortest Paths

one-to-all
- ↳ BFS usage
- ↳ Dijkstra
- ↳ Bellman-Ford
- ↳ negative-closed w. detection

all-to-all
- ↳ One-to-all usage
- ↳ Floyd-Warshall
- ↳ Johnson
- ↳ #walks using A_G

↳ overview

MST
- ↳ Prim
- ↳ Boruvka
- ↳ Kruskal

Outline

- Quiz
- Exercise Sheets
- BFS
- Code Expert - Graph Sets

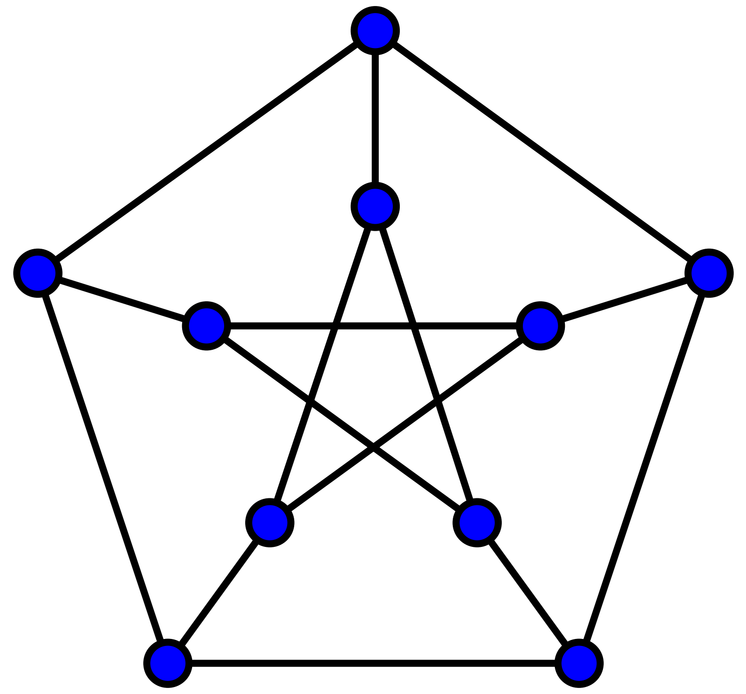
Quiz

Exercise Sheet 8

Bonus Feedback

- 8.1 : If it says translate the conditions, translate the conditions ! 👏🏻
- 8.5 : 👏🏻👏🏻👏🏻
- For all v ∃ hamiltonian path starting with v $\not\Rightarrow$ There is a hamiltonian cycle

  Example: Petersen Graph



- length of a path = #edges
  - n vertices in a path means that the path has a length of n -1

Peergrading and rest

- Exercise Sheet 9 peergrading
  - 9.4 this week
  - Emails will be sent
- New groups for sheet 10 !
- Sheet 7 grades coming up shortly

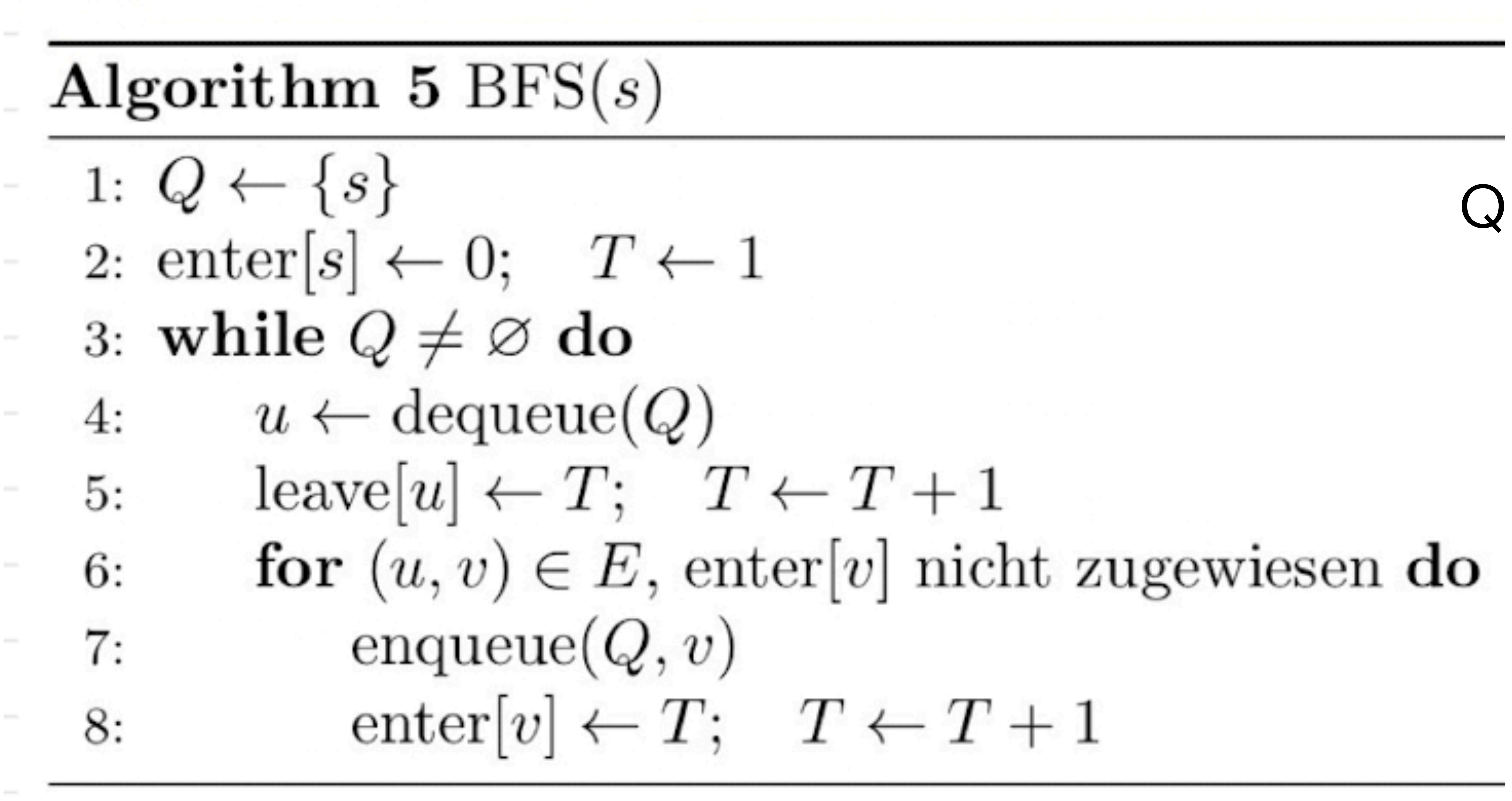
Graph Searches

BFS

Graph Searches

BFS - with pre and post order

Runtime : O (|V| + |E|)



Algorithm 5 BFS(s)

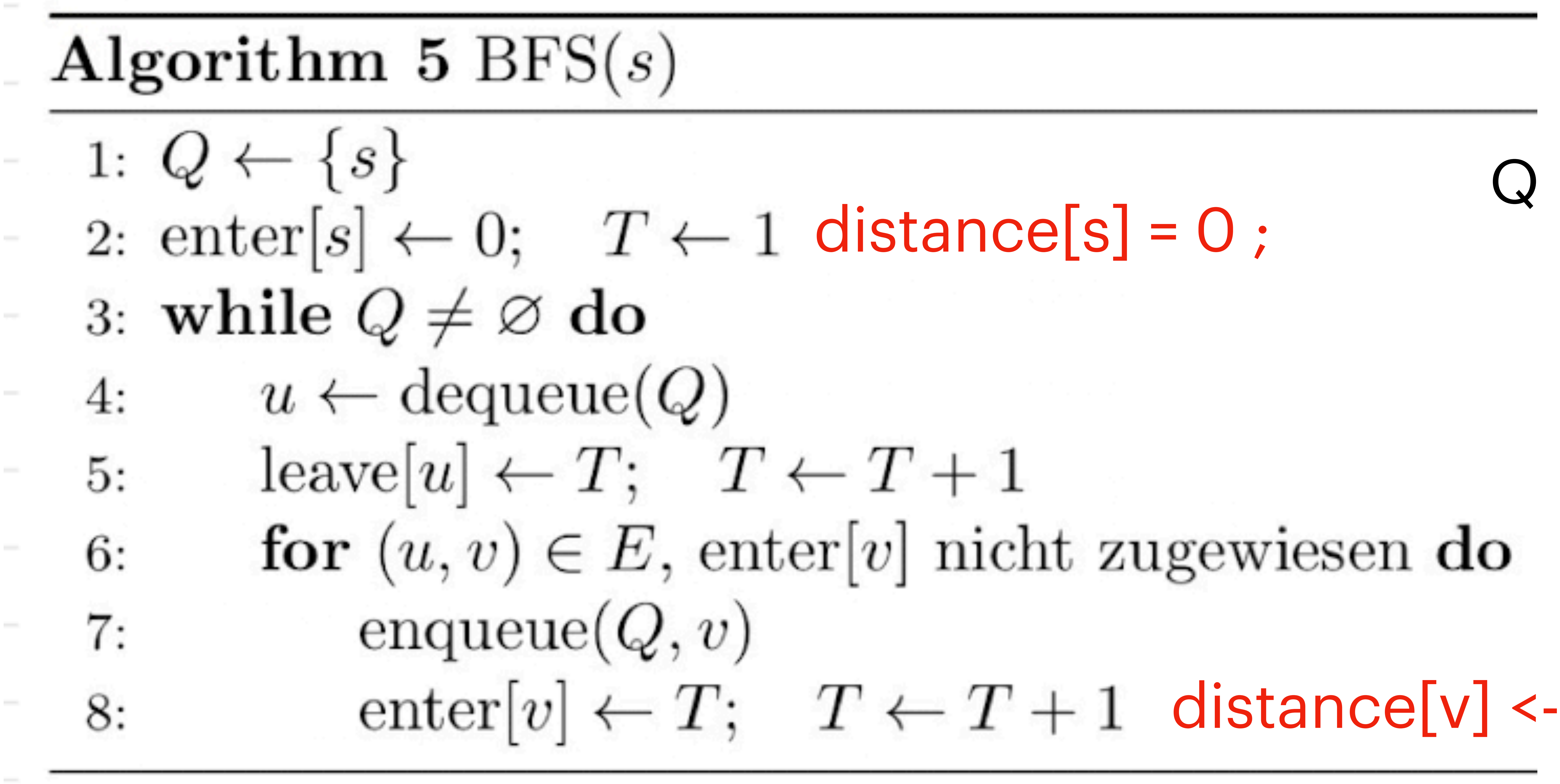
```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

Q is a FIFO queue

Graph Searchs

BFS - with pre and post order + distances

Runtime : O (|V| + |E|)



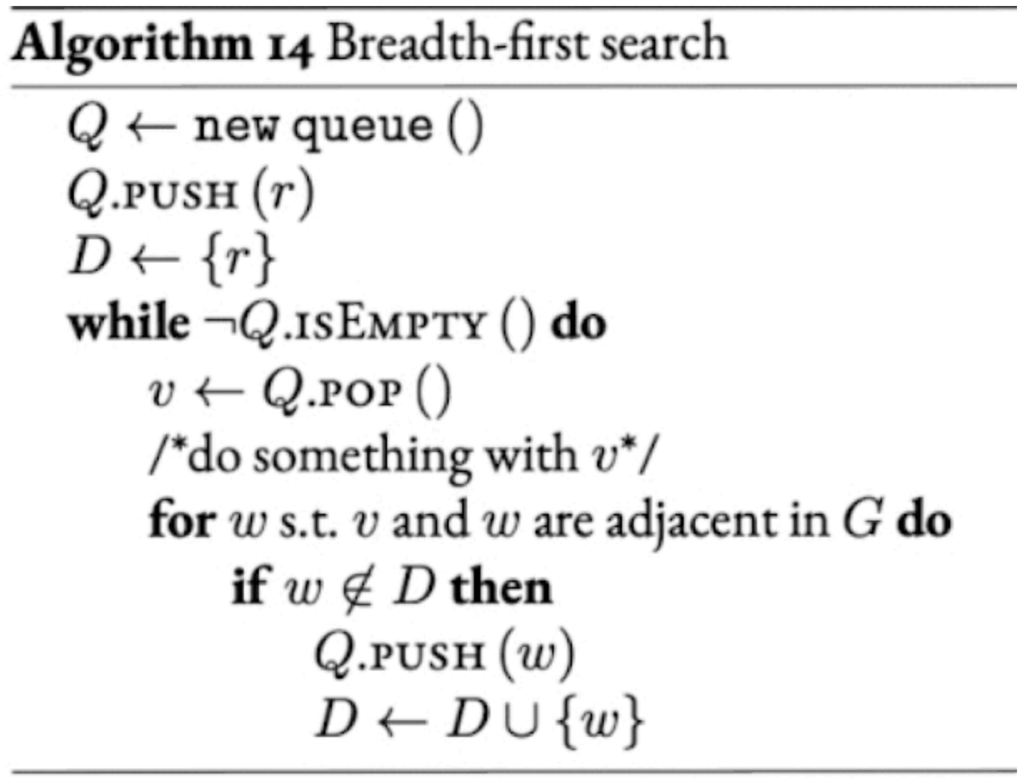
Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;   $T \leftarrow 1$   distance[s] = 0 ;
**while** $Q \neq \emptyset$ **do**
    $u \leftarrow$ dequeue$(Q)$
    leave$[u] \leftarrow T$;   $T \leftarrow T + 1$
    **for** $(u, v) \in E$, enter$[v]$ nicht zugewiesen **do**
        enqueue$(Q, v)$
        enter$[v] \leftarrow T$;   $T \leftarrow T + 1$   distance[v] <- distance[u] + 1 ;
```

Q is a FIFO queue

Graph Searches

BFS - with pre and post order + distances

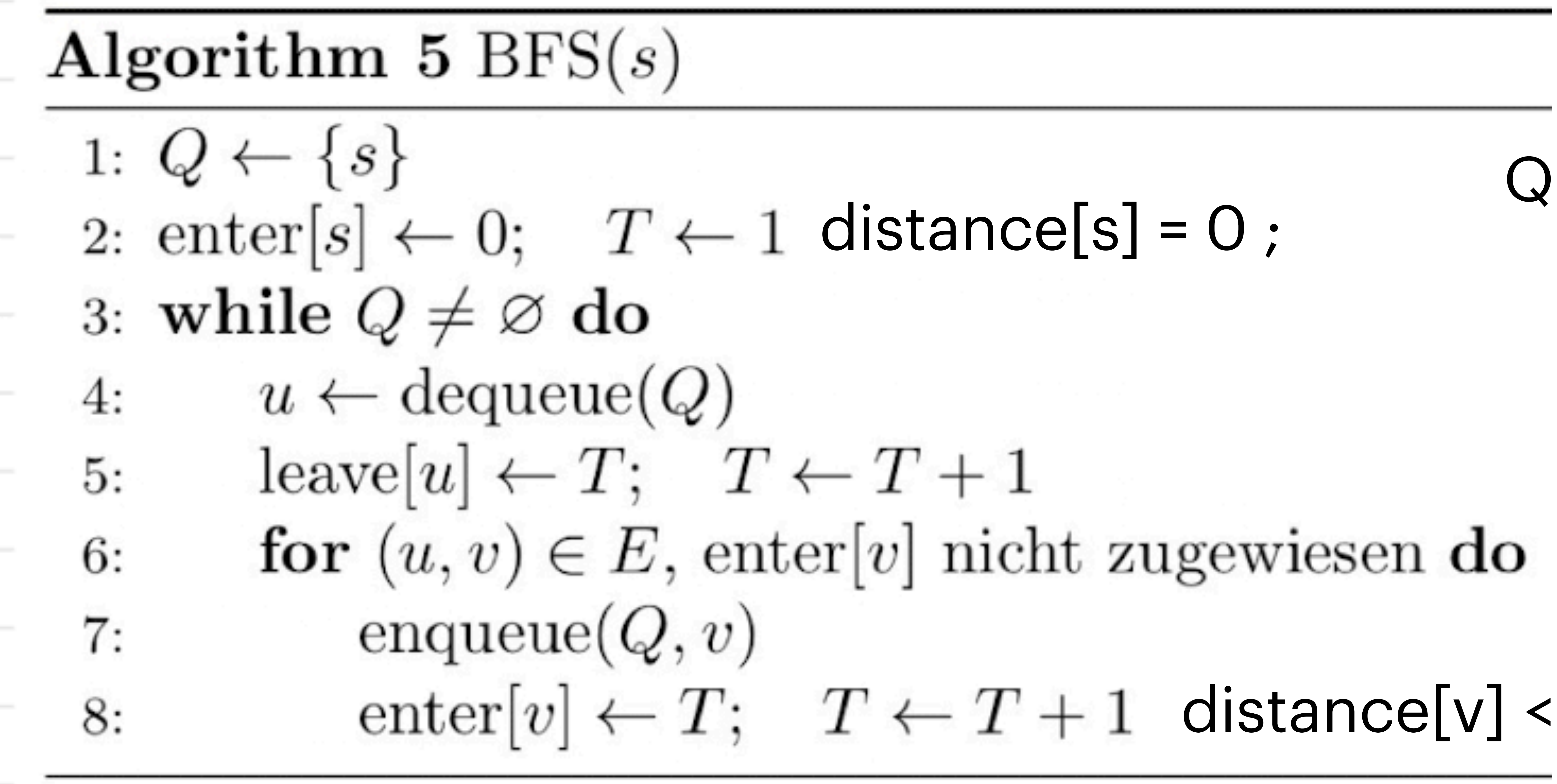


Algorithm 14 Breadth-first search

```
$Q \leftarrow$ new queue ()
$Q$.PUSH $(r)$
$D \leftarrow \{r\}$
while $\neg Q$.ISEMPTY () do
    $v \leftarrow Q$.POP ()
    /*do something with $v$*/
    for $w$ s.t. $v$ and $w$ are adjacent in $G$ do
        if $w \notin D$ then
            $Q$.PUSH $(w)$
            $D \leftarrow D \cup \{w\}$
```

Runtime : O (|V| + |E|)

Q is a FIFO queue

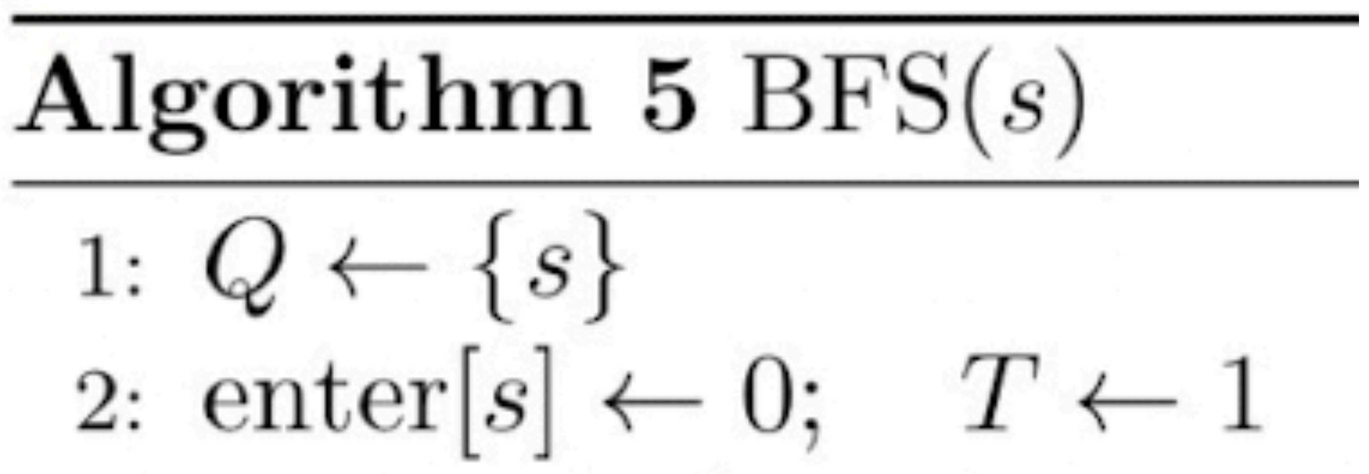


Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;    $T \leftarrow 1$    distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue$(Q)$
    leave$[u] \leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter$[v]$ nicht zugewiesen do
        enqueue$(Q, v)$
        enter$[v] \leftarrow T$;    $T \leftarrow T + 1$    distance[v] <- distance[u] + 1 ;
```

Graph Searchs

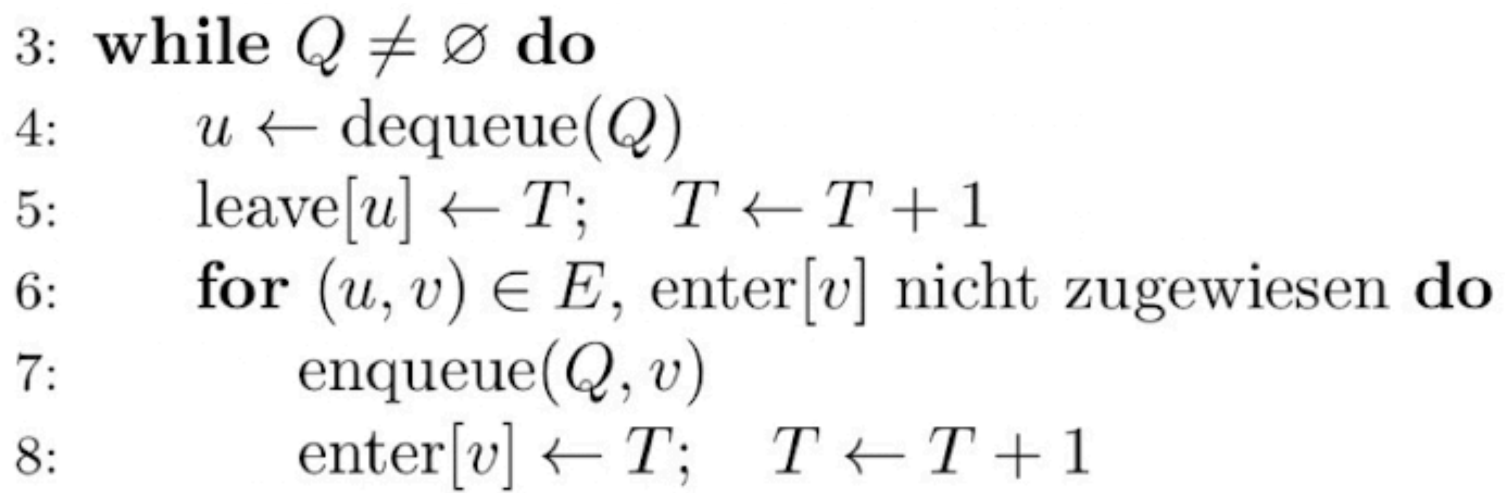
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;   $T \leftarrow 1$
```

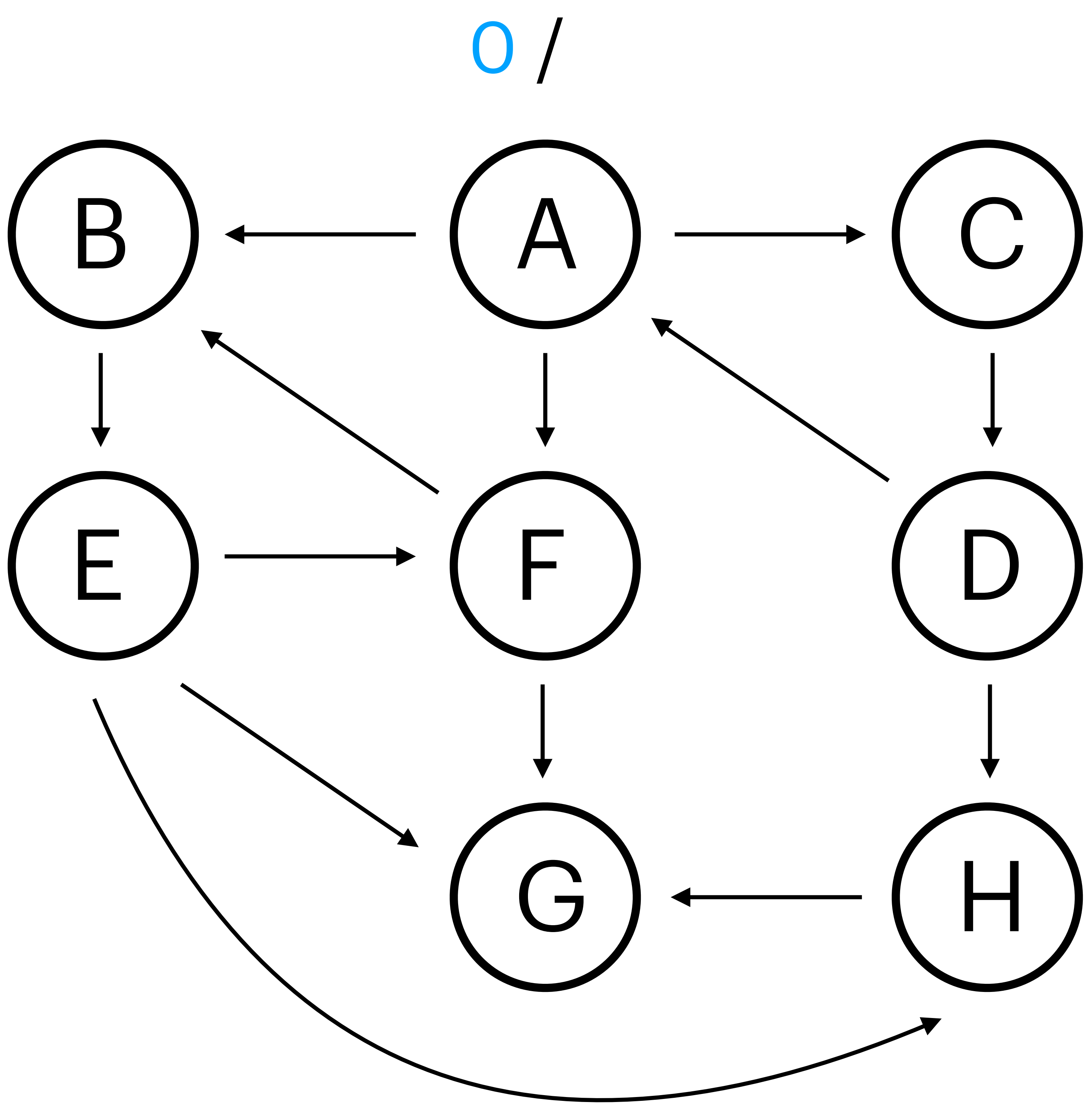
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue$(Q)$
    leave$[u] \leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter$[v]$ nicht zugewiesen do
        enqueue$(Q, v)$
        enter$[v] \leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : A



enter[] :

A	B	C	D	E	F	G	H
0							

leave[] :

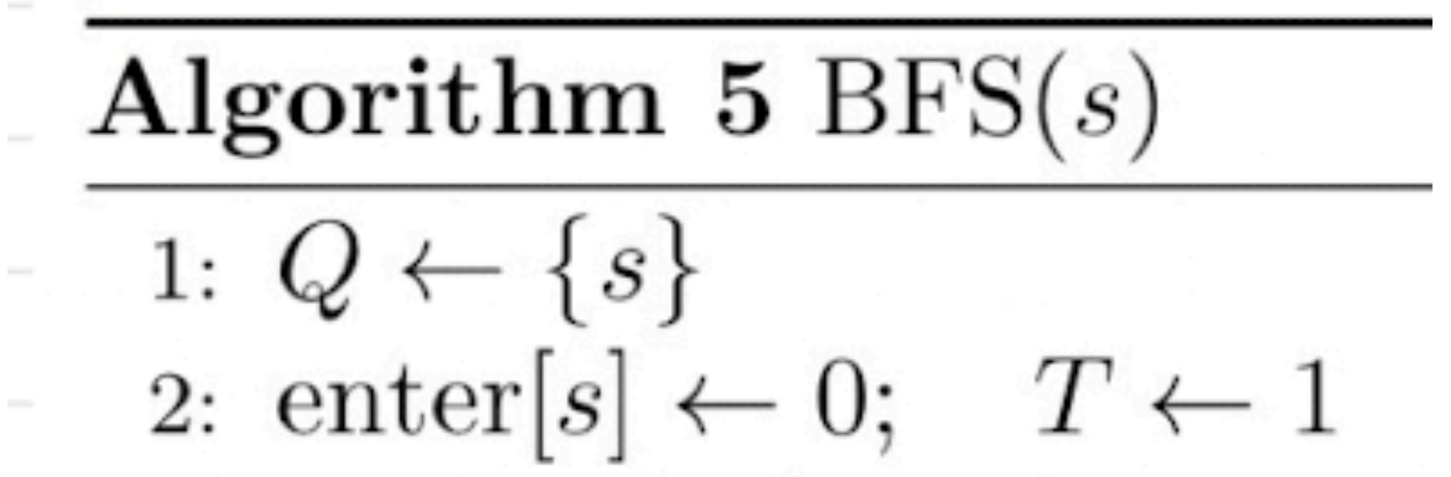
A	B	C	D	E	F	G	H

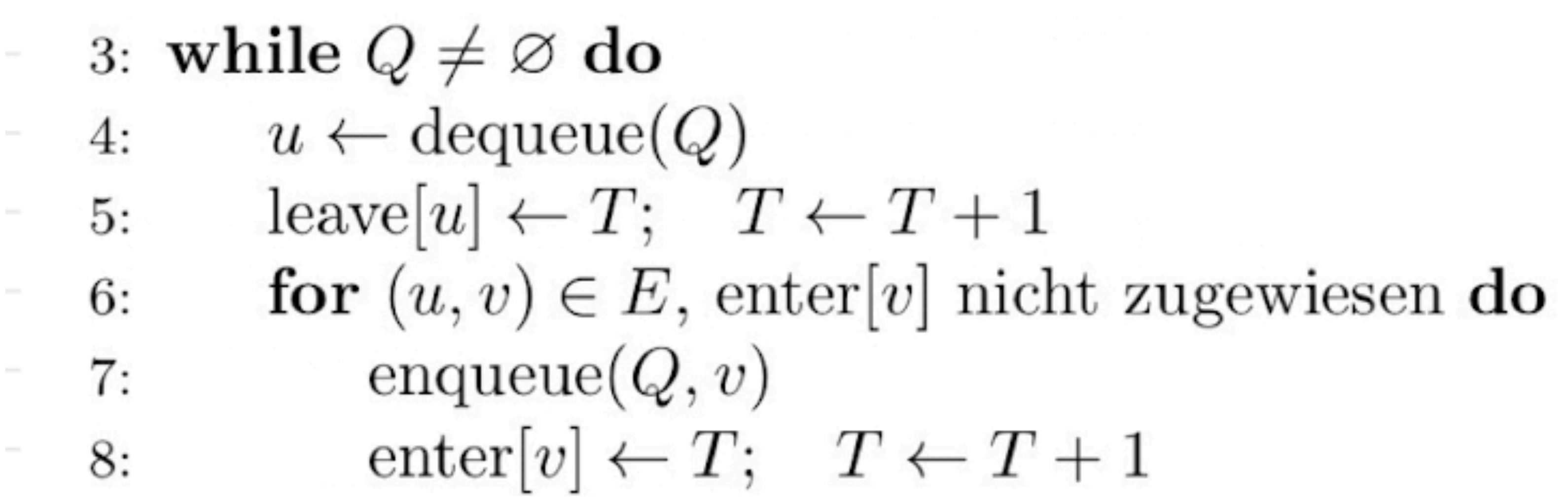
distance[] :

A	B	C	D	E	F	G	H
0							

Graph Searchs

BFS - Example

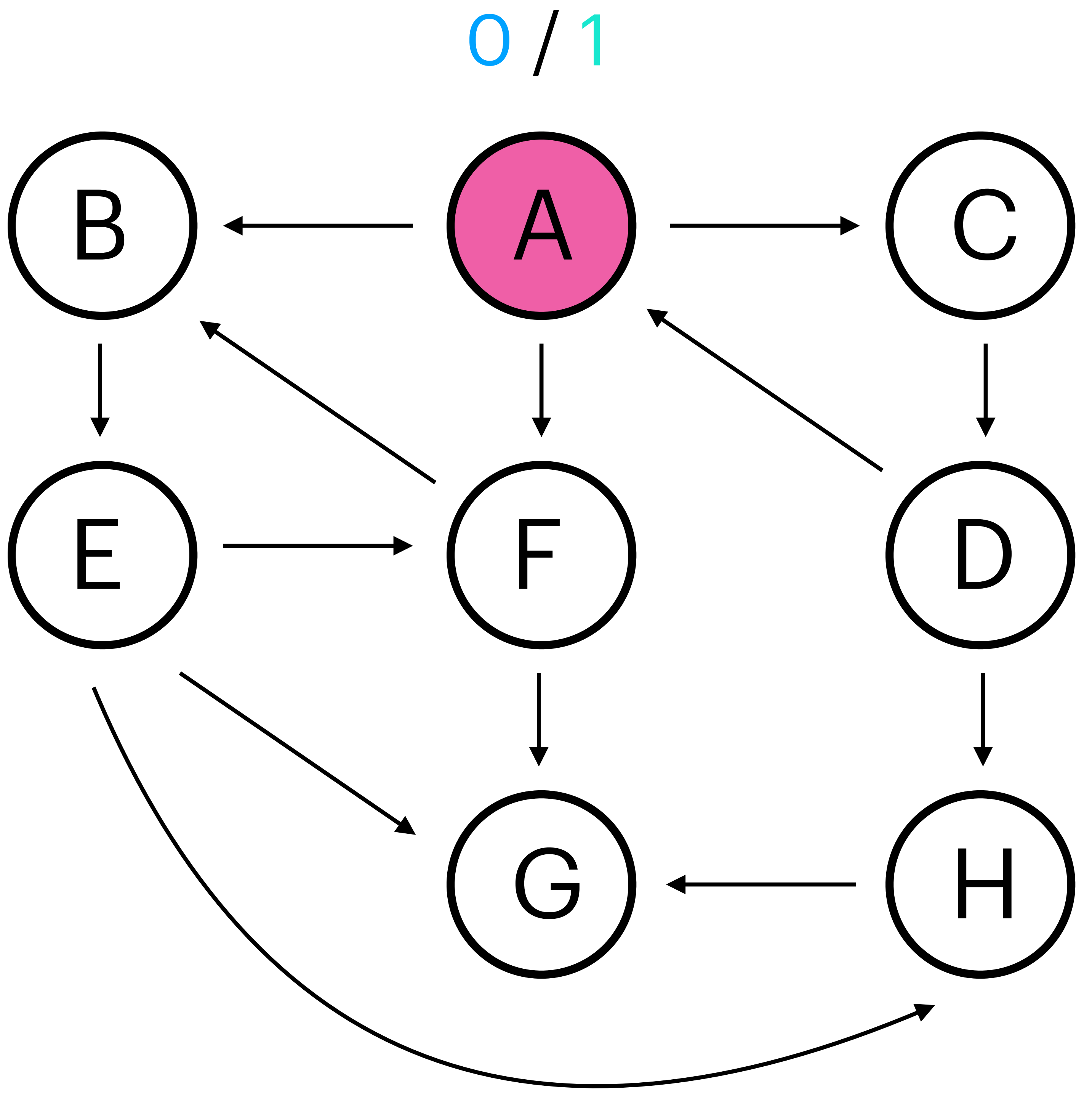




```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q :

u = A



enter[] :

A	B	C	D	E	F	G	H
0							

leave[] :

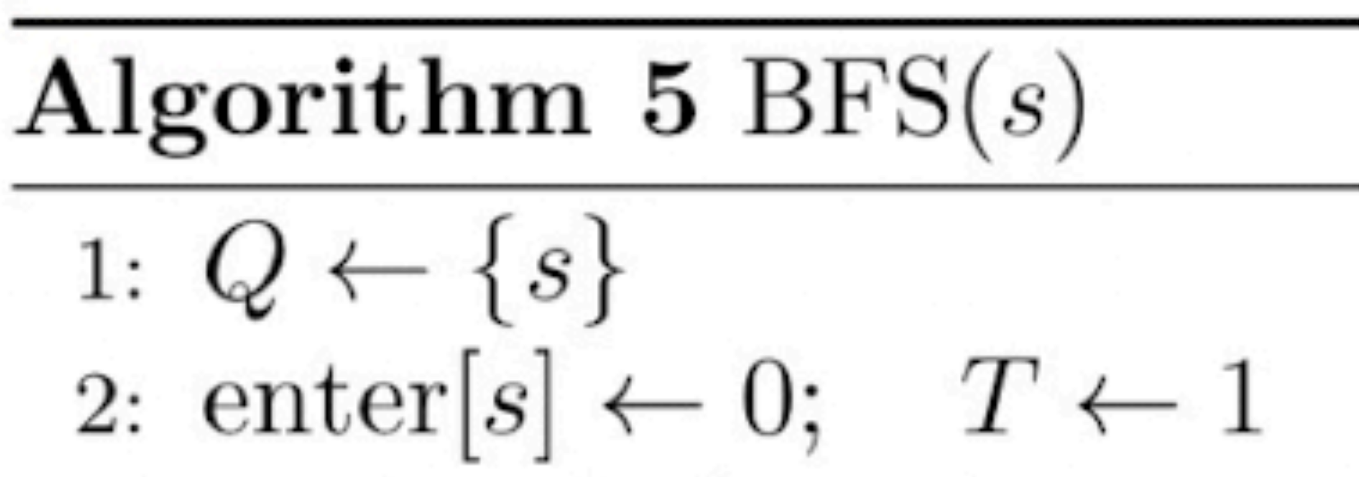
A	B	C	D	E	F	G	H
1							

distance[] :

A	B	C	D	E	F	G	H
0							

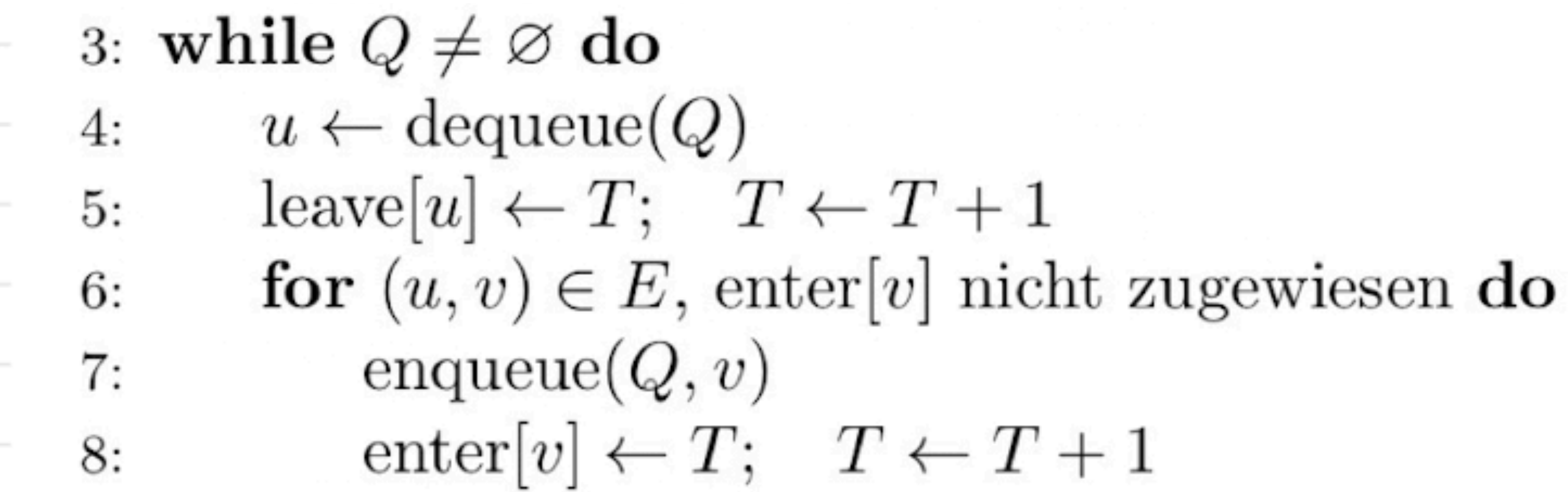
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

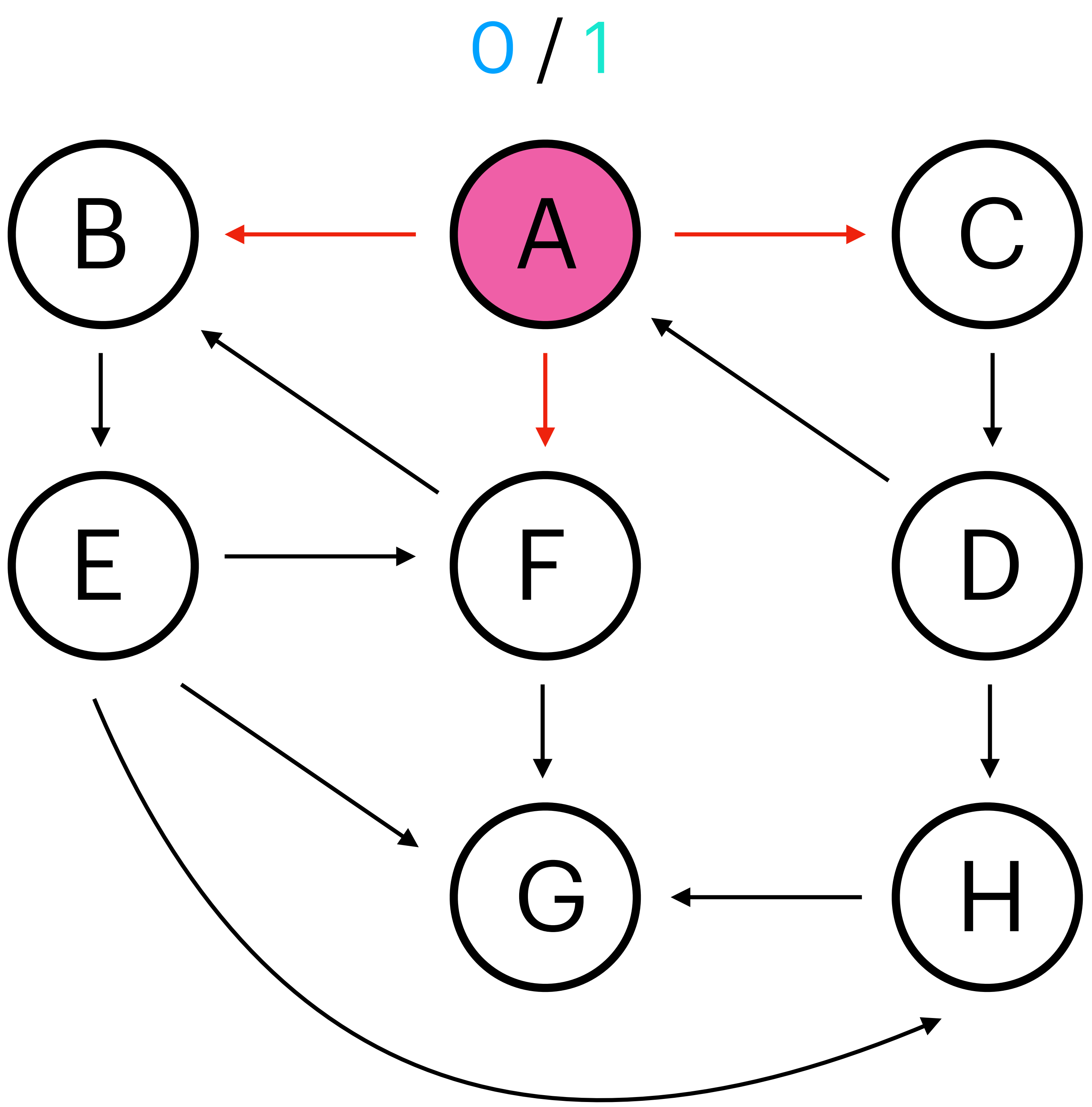


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q :

u = A



enter[] :

A	B	C	D	E	F	G	H
0							

leave[] :

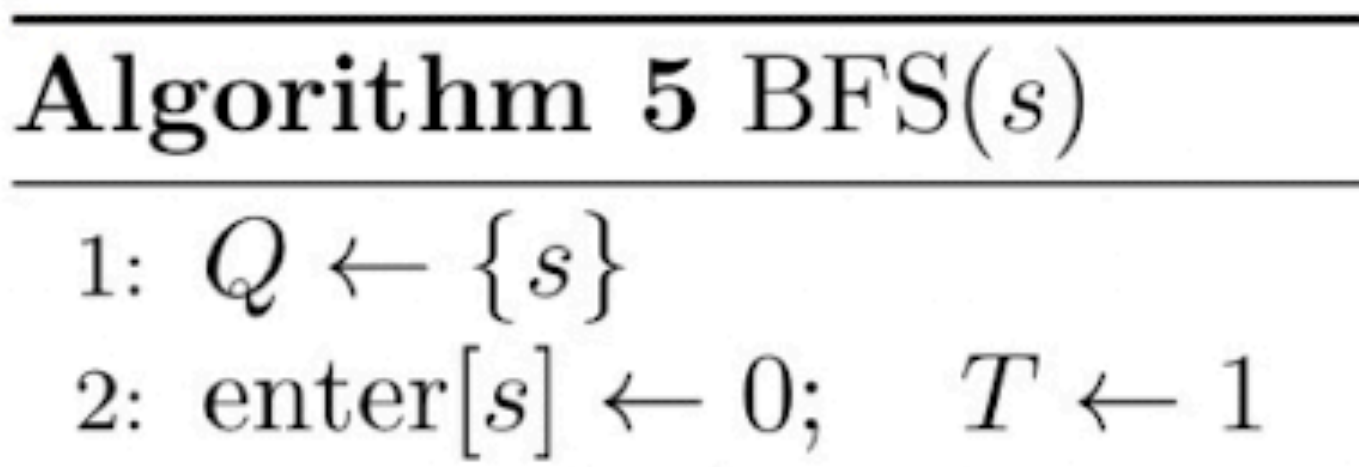
A	B	C	D	E	F	G	H
1							

distance[] :

A	B	C	D	E	F	G	H
0							

Graph Searchs

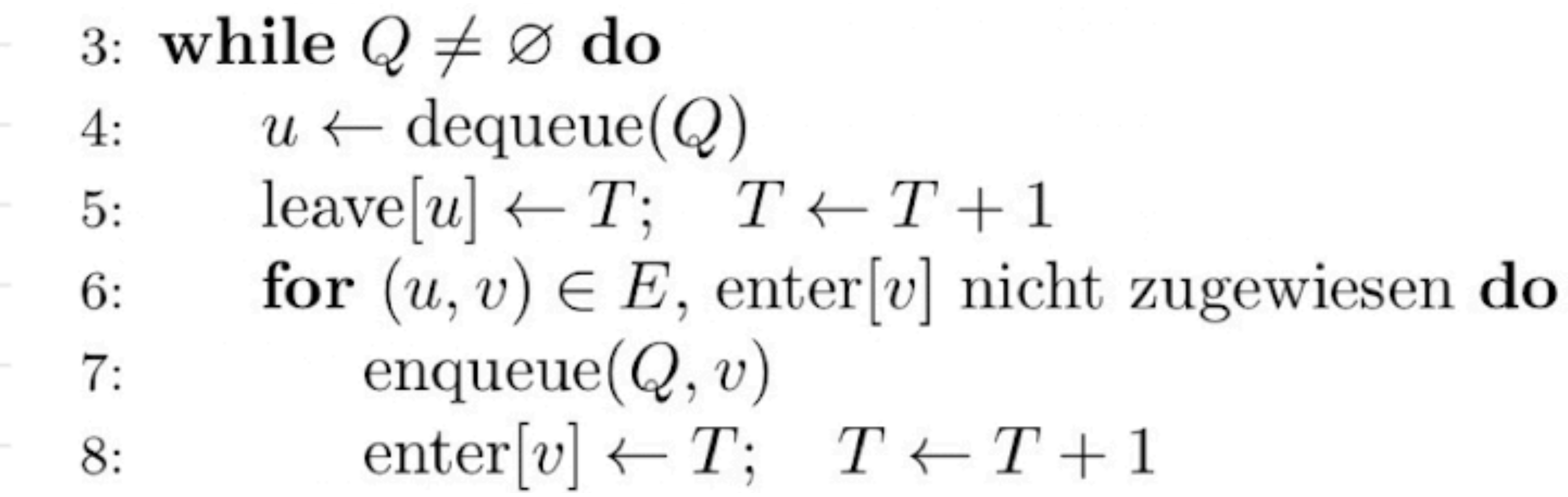
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

distance[s] = 0 ;

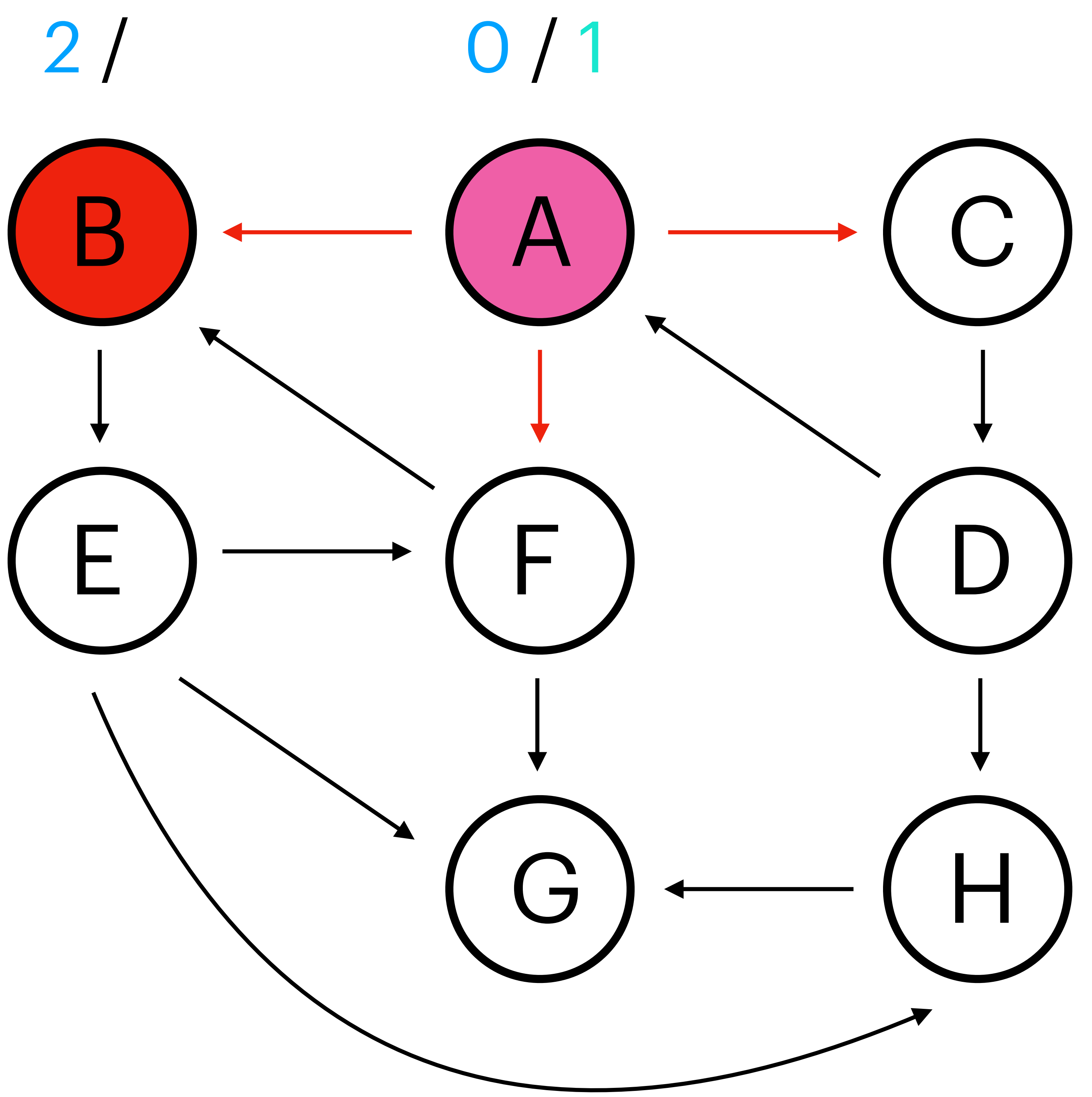


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : B

u = A



enter[] :

A	B	C	D	E	F	G	H
0	2						

leave[] :

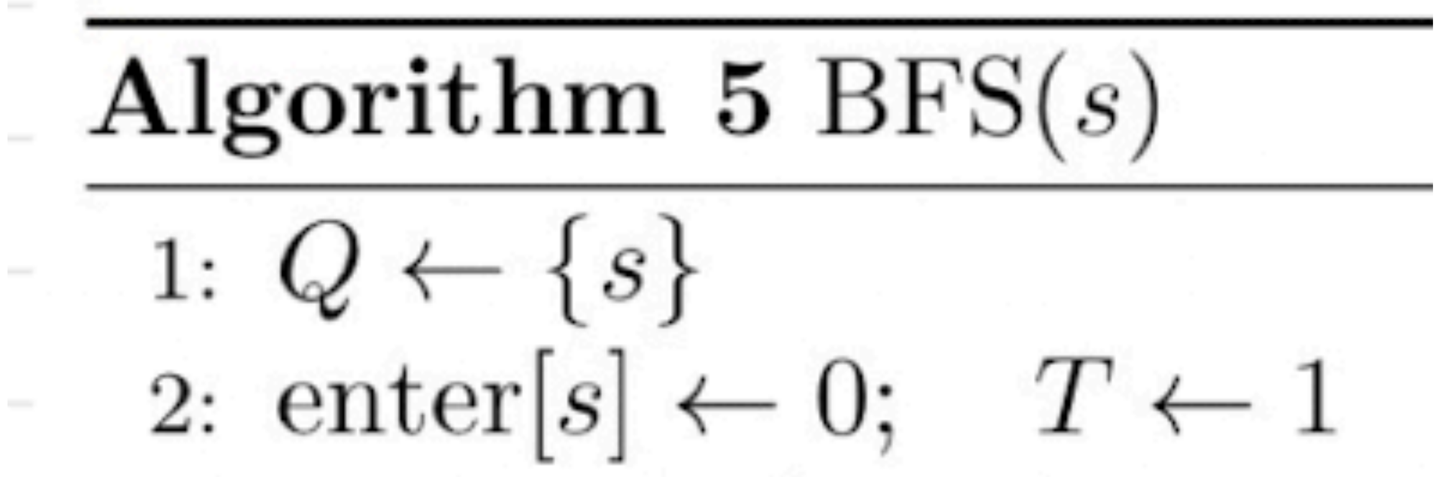
A	B	C	D	E	F	G	H
1							

distance[] :

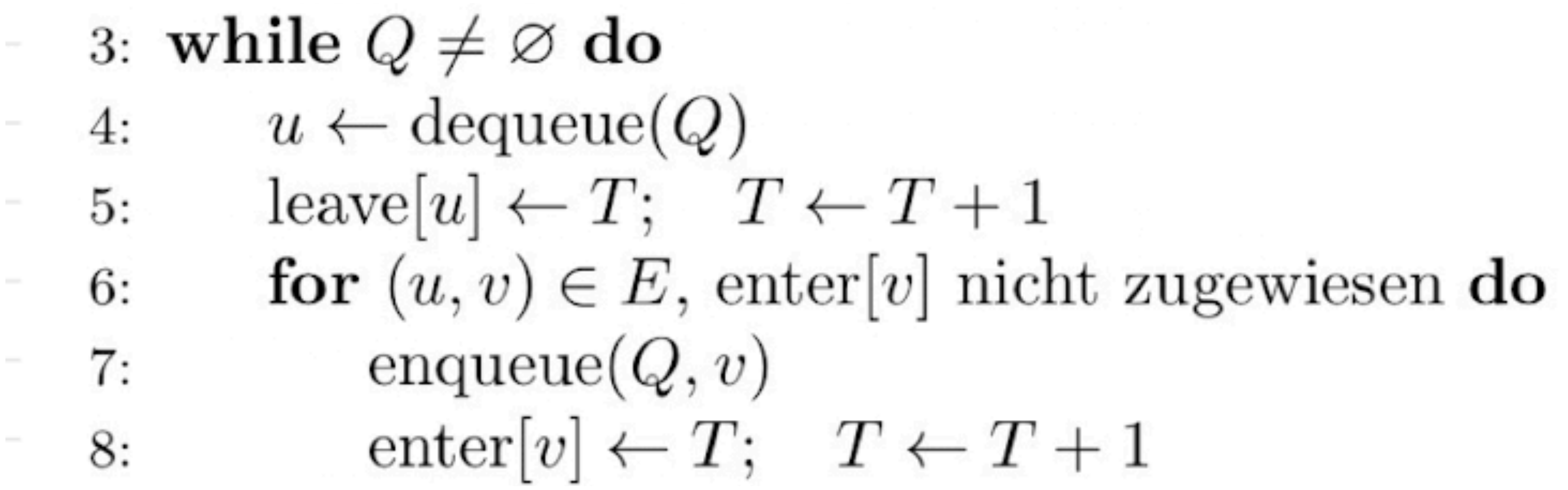
A	B	C	D	E	F	G	H
0	1						

Graph Searchs

BFS - Example



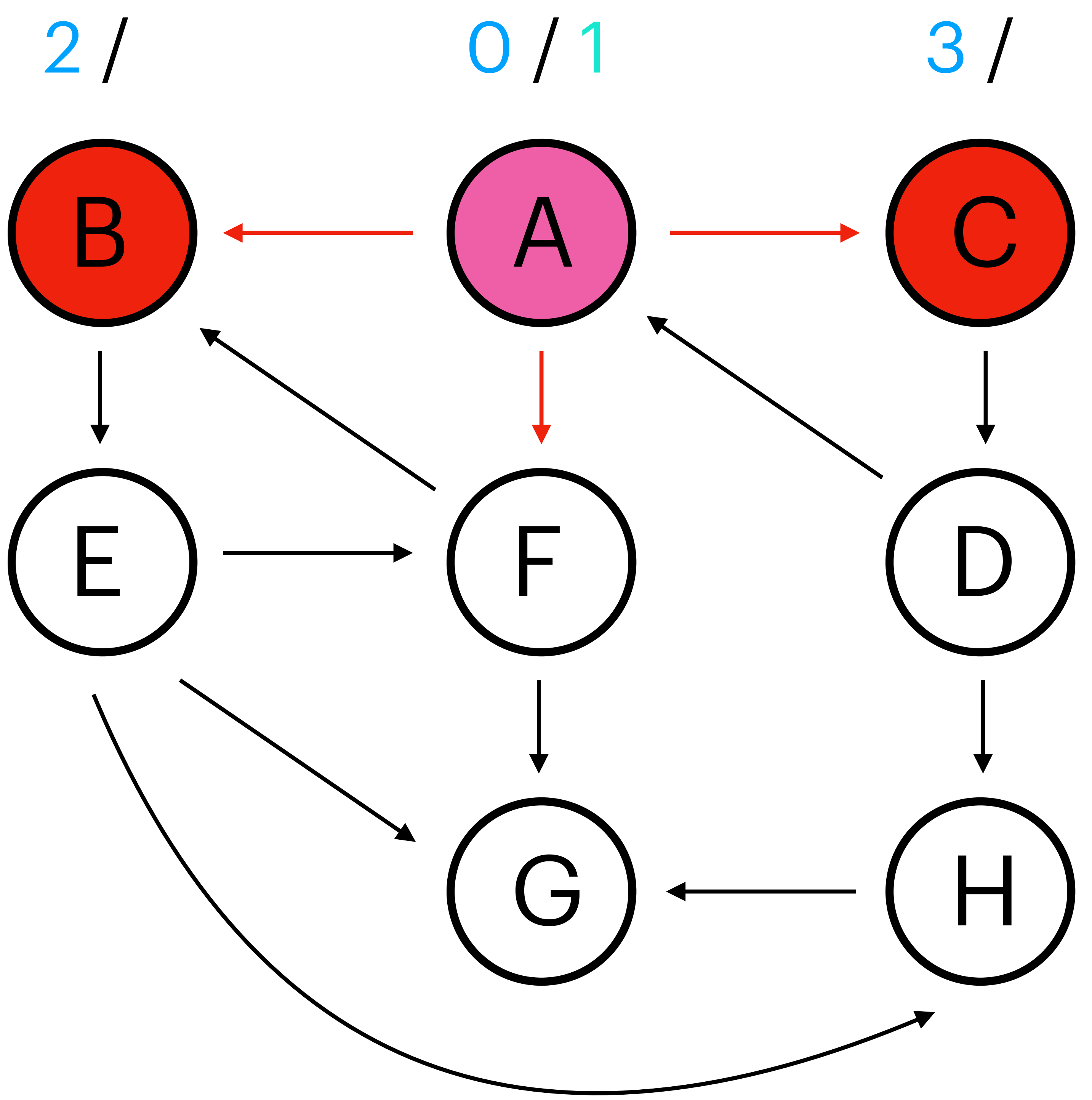
```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : B - C

u = A



enter[] :

A	B	C	D	E	F	G	H
0	2	3					

leave[] :

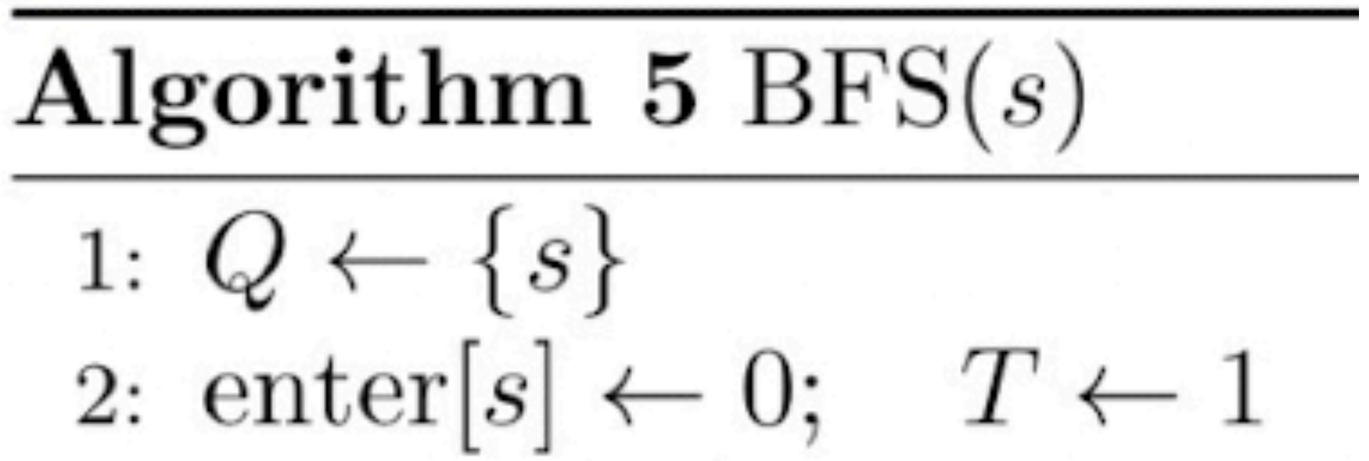
A	B	C	D	E	F	G	H
1							

distance[] :

A	B	C	D	E	F	G	H
0	1	1					

Graph Searchs

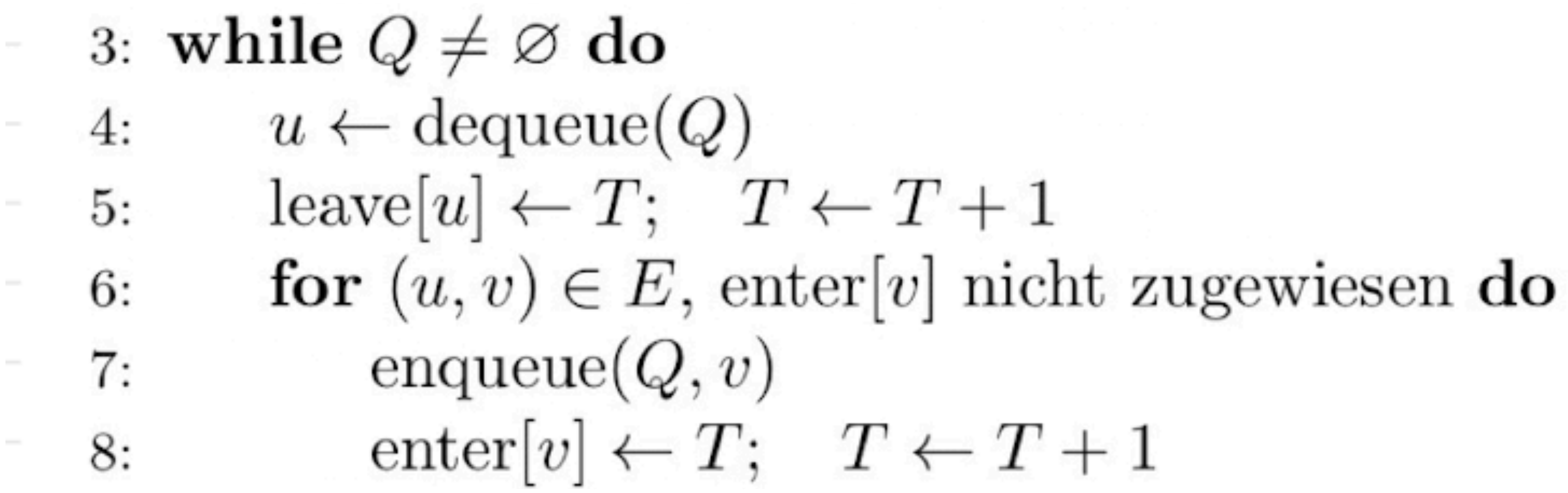
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;



```
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue$(Q)$
    leave$[u] \leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter$[v]$ nicht zugewiesen do
        enqueue$(Q, v)$
        enter$[v] \leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : B - C - F

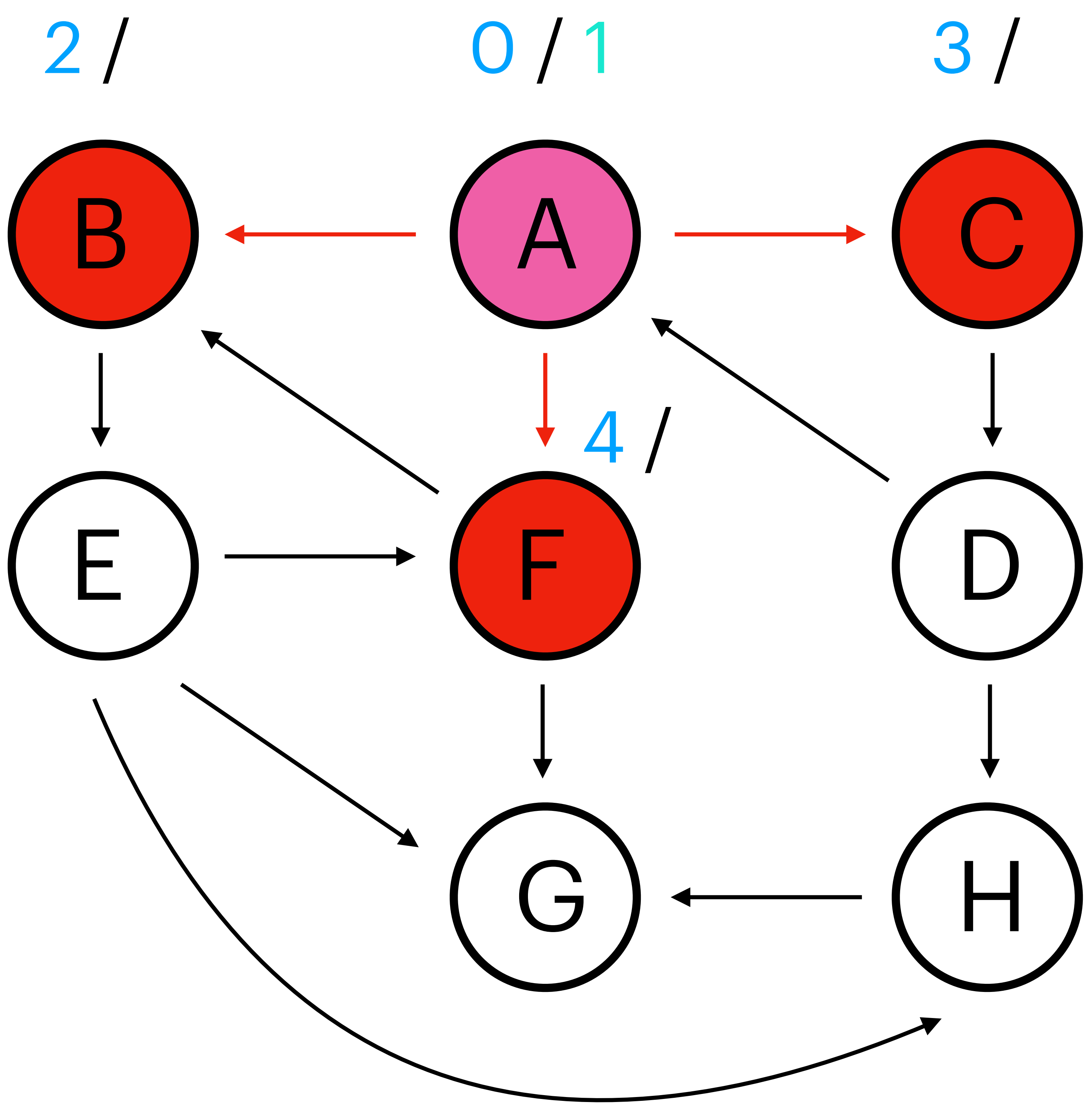
u = A

enter[] :

A	B	C	D	E	F	G	H
0	2	3			4		

leave[] :

A	B	C	D	E	F	G	H
1							

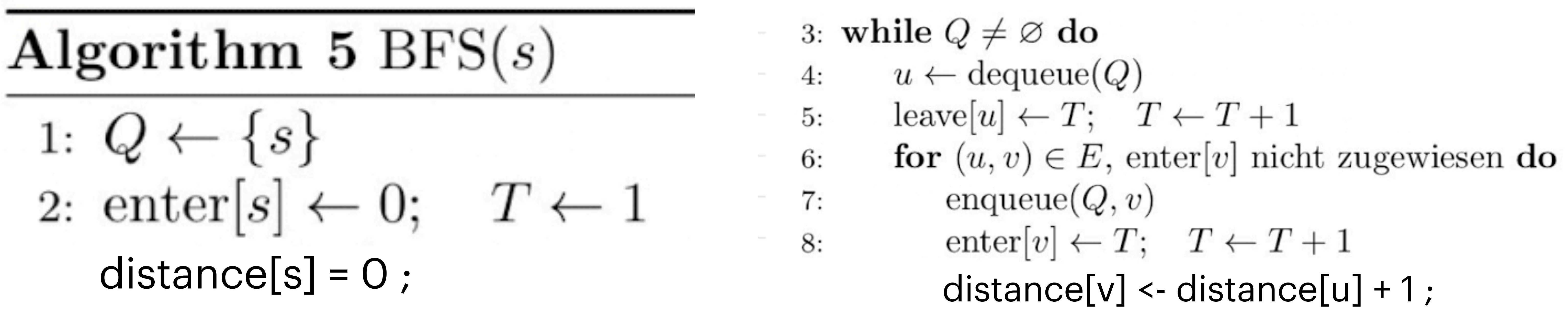


distance[] :

A	B	C	D	E	F	G	H
0	1	1			1		

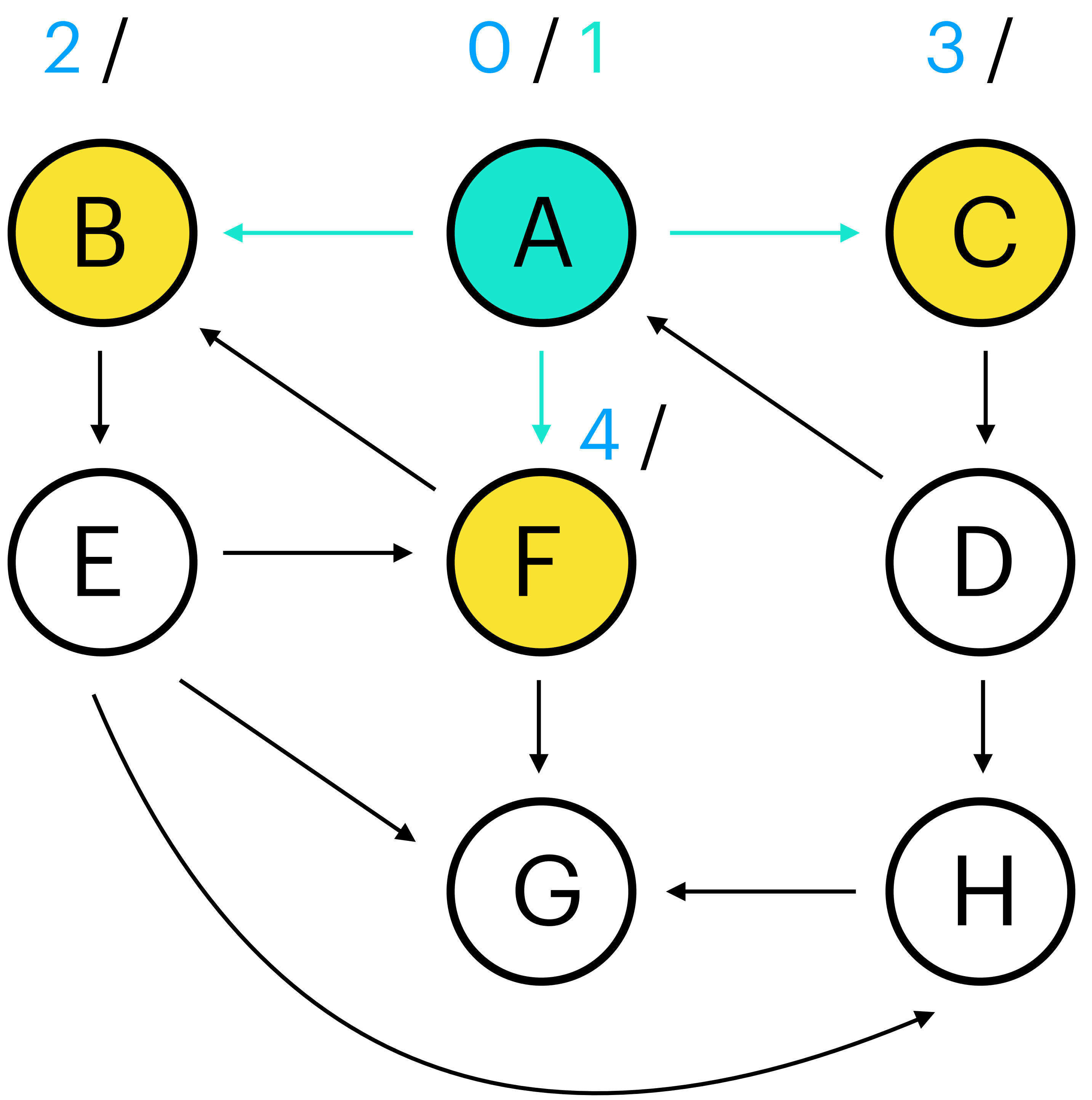
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : B - C - F



enter[] :

A	B	C	D	E	F	G	H
0	2	3			4		

leave[] :

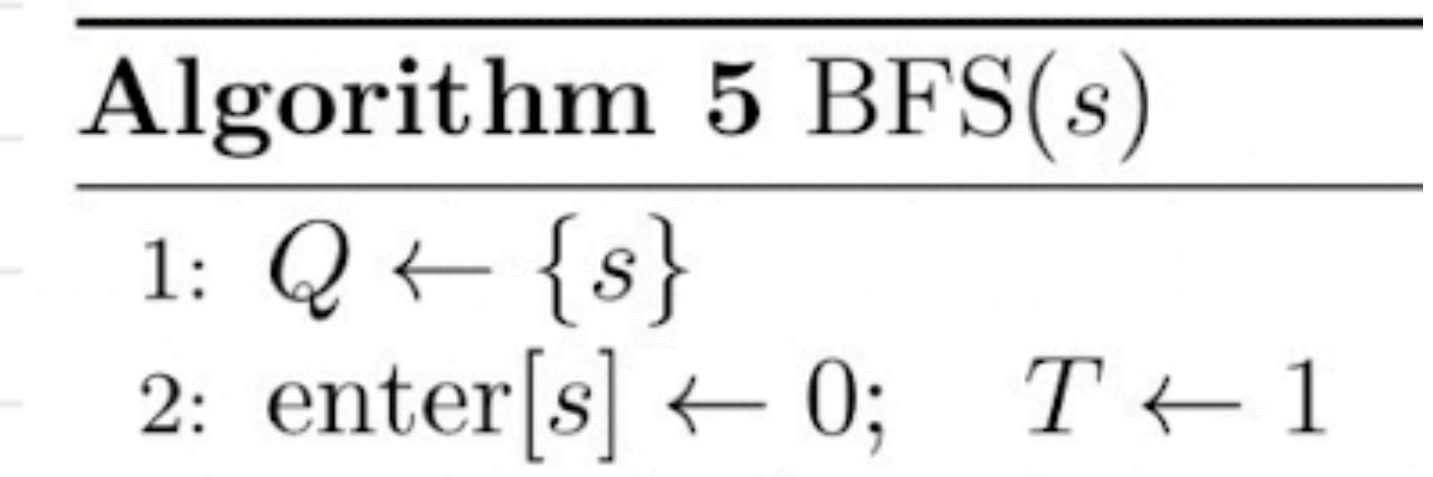
A	B	C	D	E	F	G	H
1							

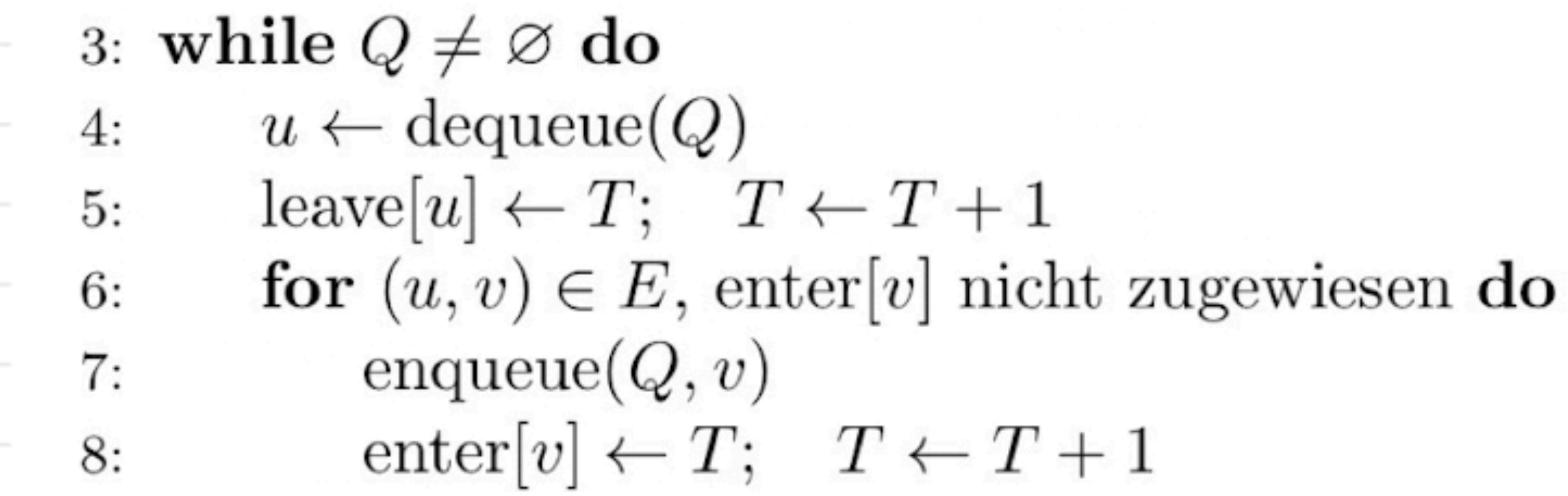
distance[] :

A	B	C	D	E	F	G	H
0	1	1			1		

Graph Searchs

BFS - Example

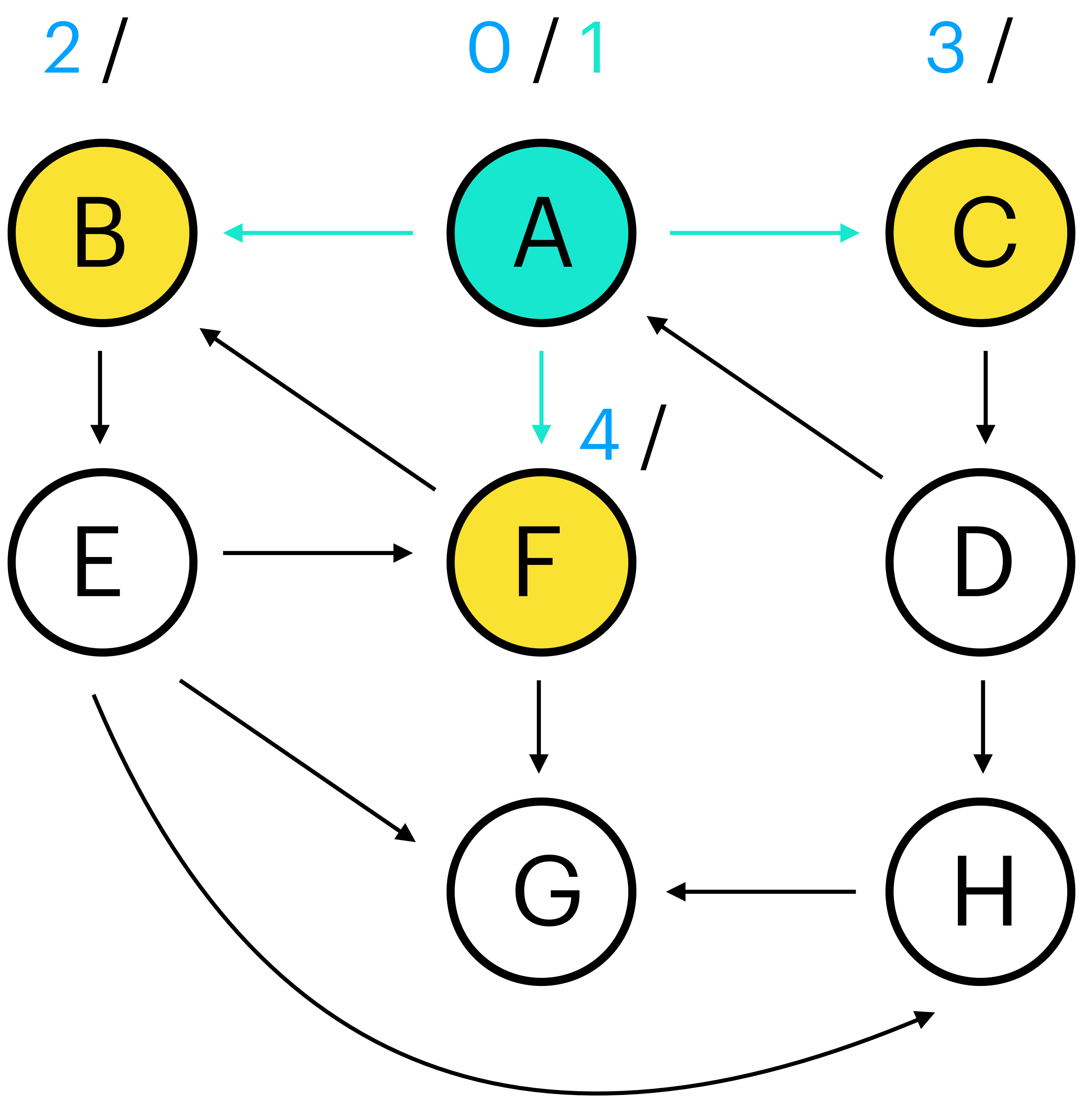




```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T+1$
    for $(u,v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T+1$
        distance[v] <- distance[u] + 1 ;
```

Q : B - C - F

FIFO



enter[] :

A	B	C	D	E	F	G	H
0	2	3			4		

leave[] :

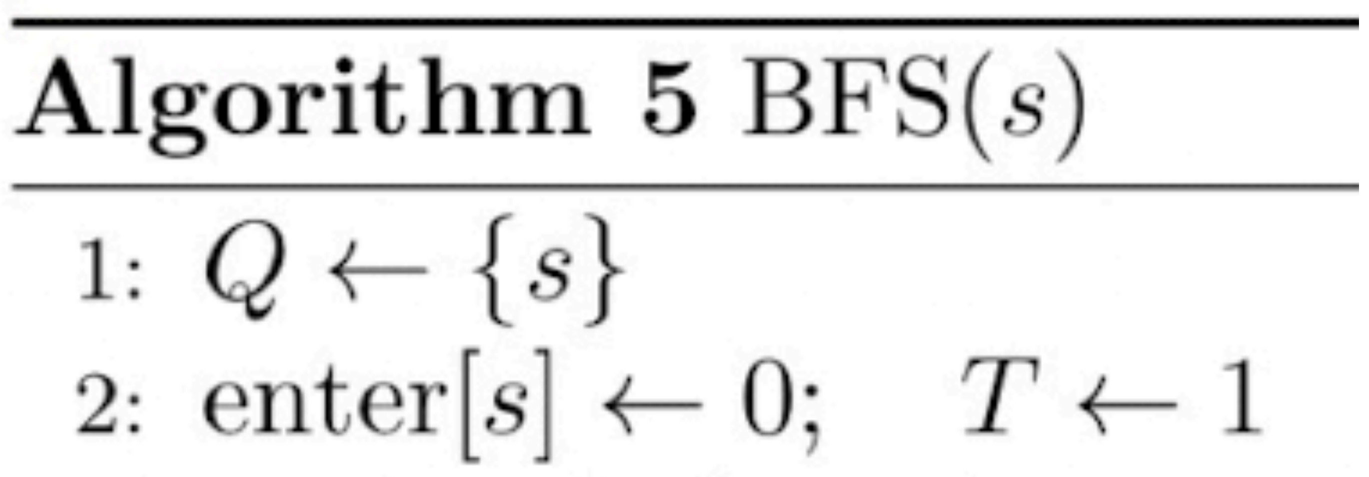
A	B	C	D	E	F	G	H
1							

distance[] :

A	B	C	D	E	F	G	H
0	1	1			1		

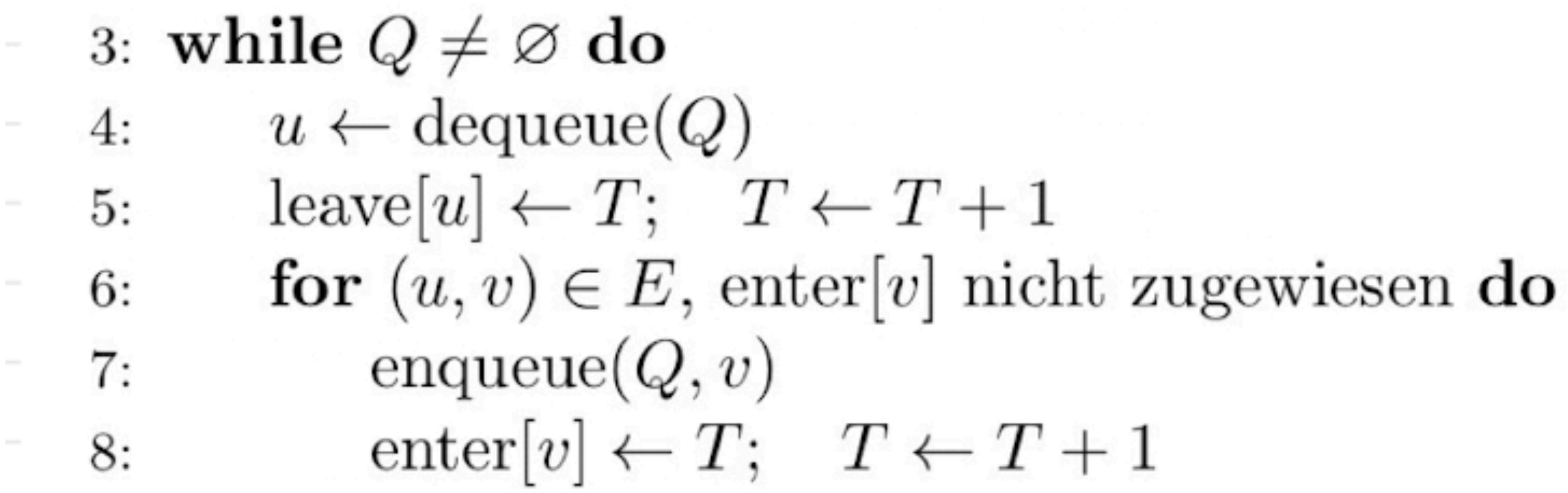
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

distance[s] = 0 ;

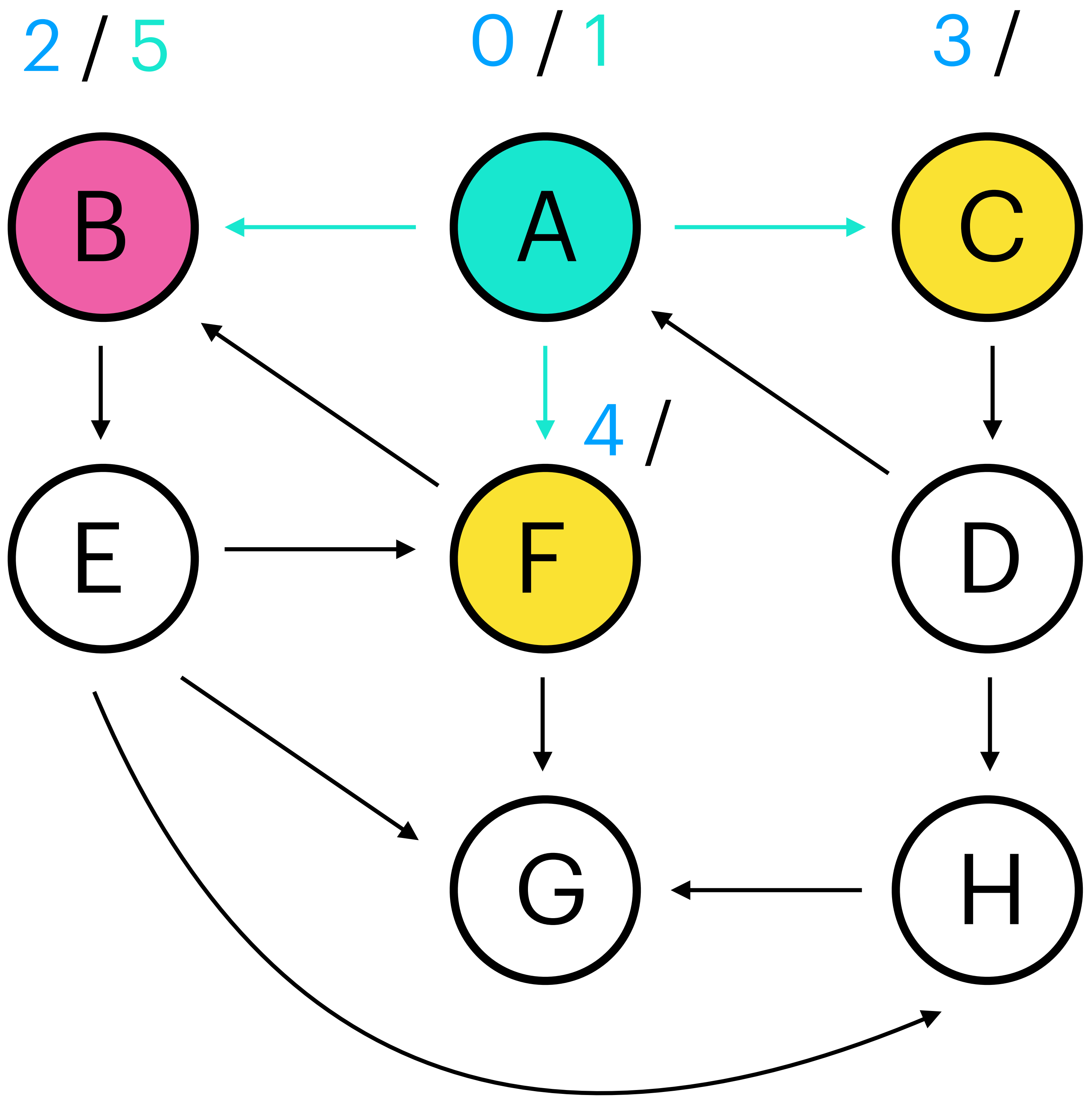


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : C - F

u = B



enter[] :

A	B	C	D	E	F	G	H
0	2	3			4		

leave[] :

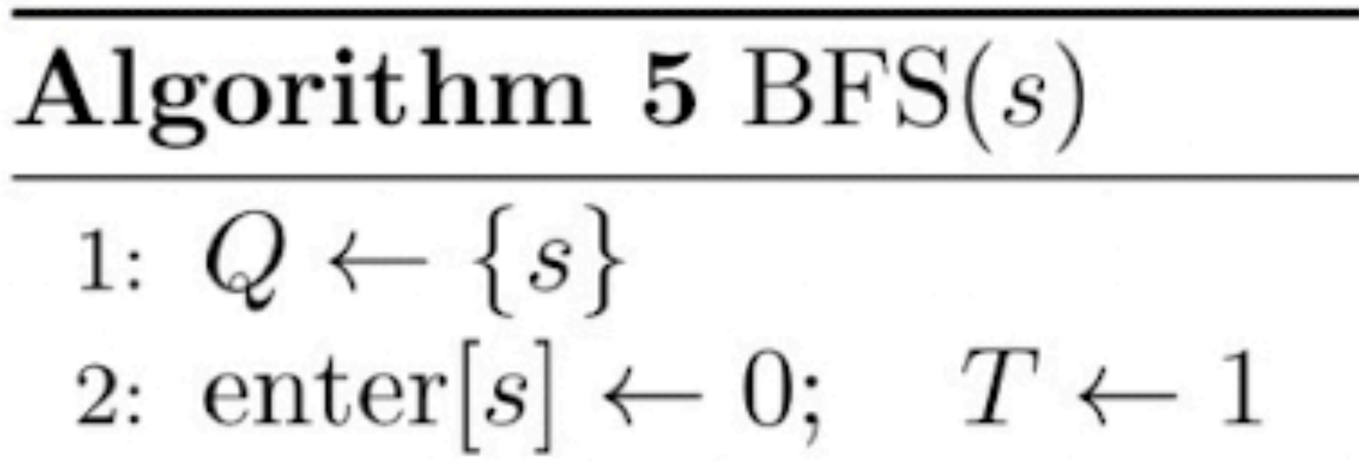
A	B	C	D	E	F	G	H
1	5						

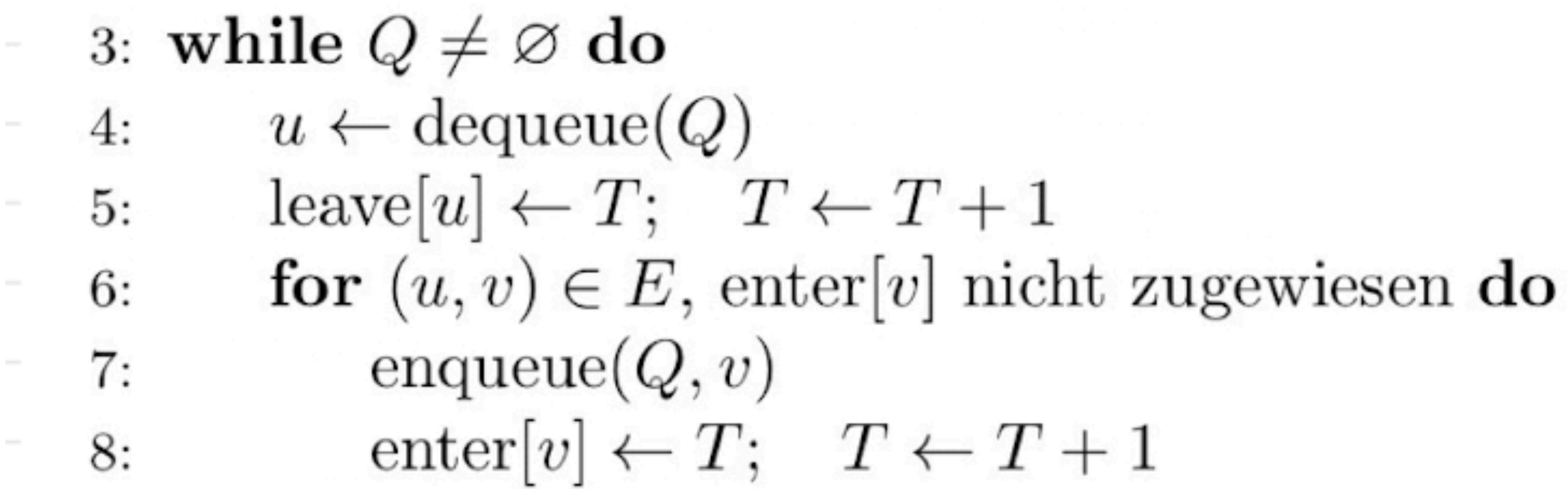
distance[] :

A	B	C	D	E	F	G	H
0	1	1			1		

Graph Searchs

BFS - Example

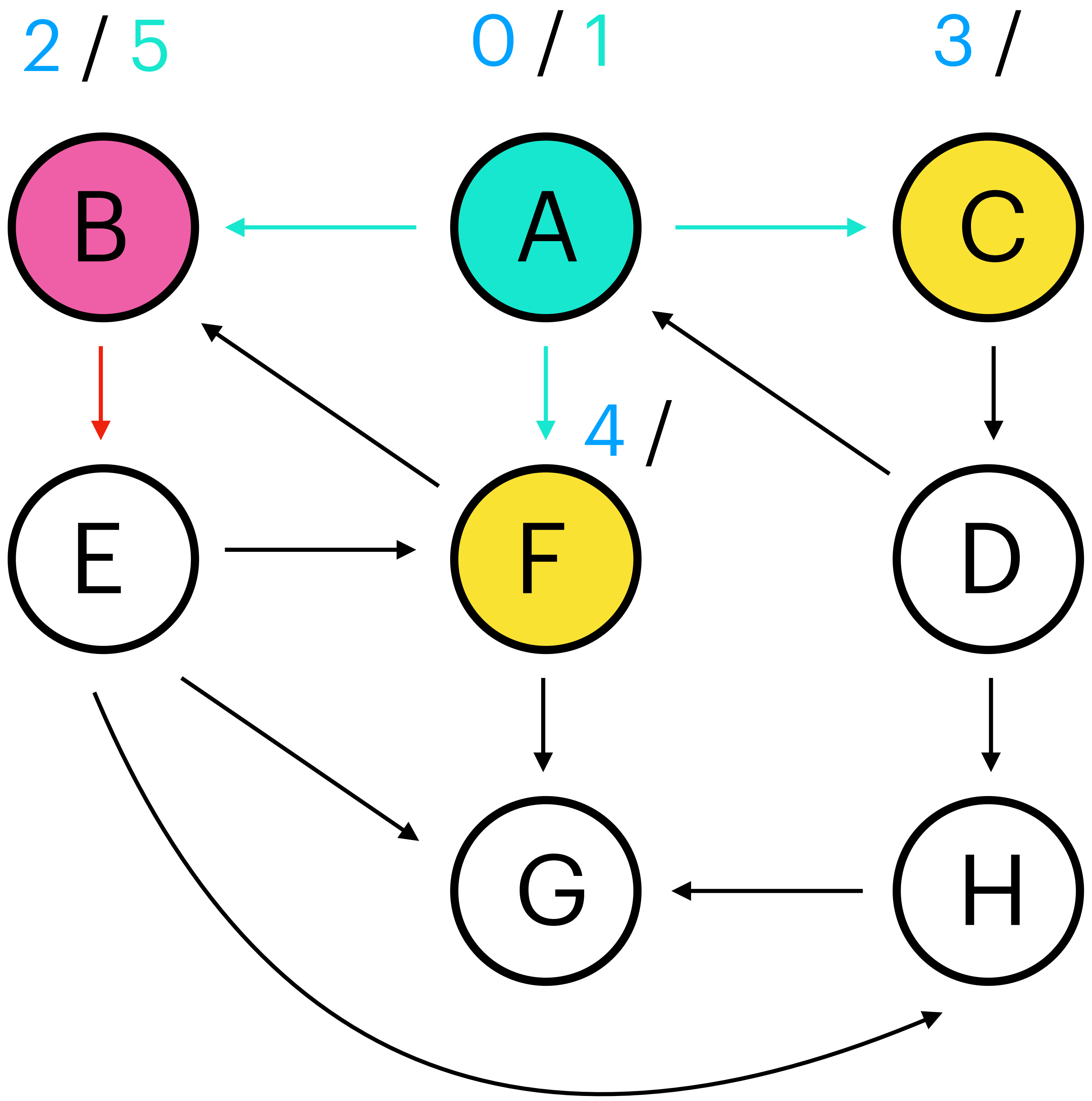




```
Algorithm 5 BFS($s$)
  $Q \leftarrow \{s\}$
  enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
  distance[s] = 0 ;
  while $Q \neq \emptyset$ do
      $u \leftarrow$ dequeue($Q$)
      leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
      for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
          enqueue($Q, v$)
          enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
          distance[v] <- distance[u] + 1 ;
```

Q : C - F

u = B



enter[] :

A	B	C	D	E	F	G	H
0	2	3			4		

leave[] :

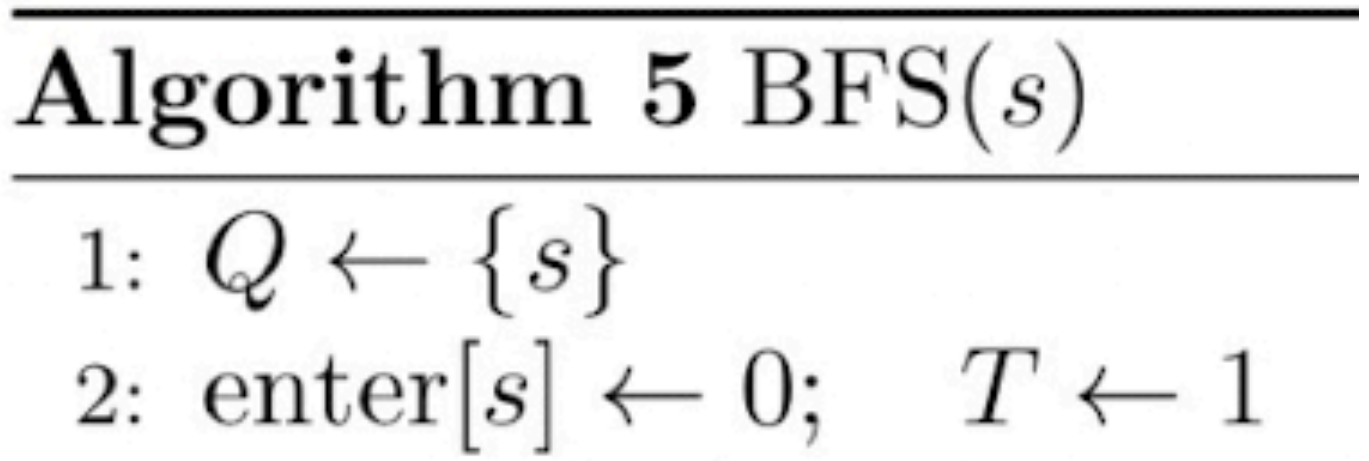
A	B	C	D	E	F	G	H
1	5						

distance[] :

A	B	C	D	E	F	G	H
0	1	1			1		

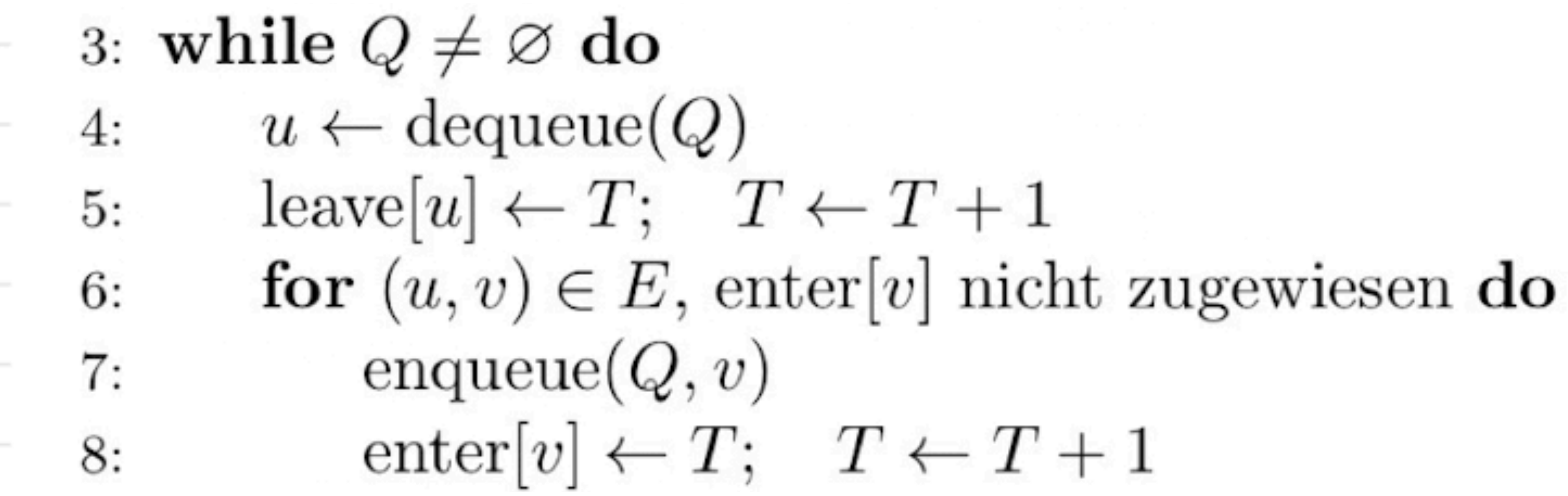
Graph Searchs

BFS - Example



```
Algorithm 5 BFS(s)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

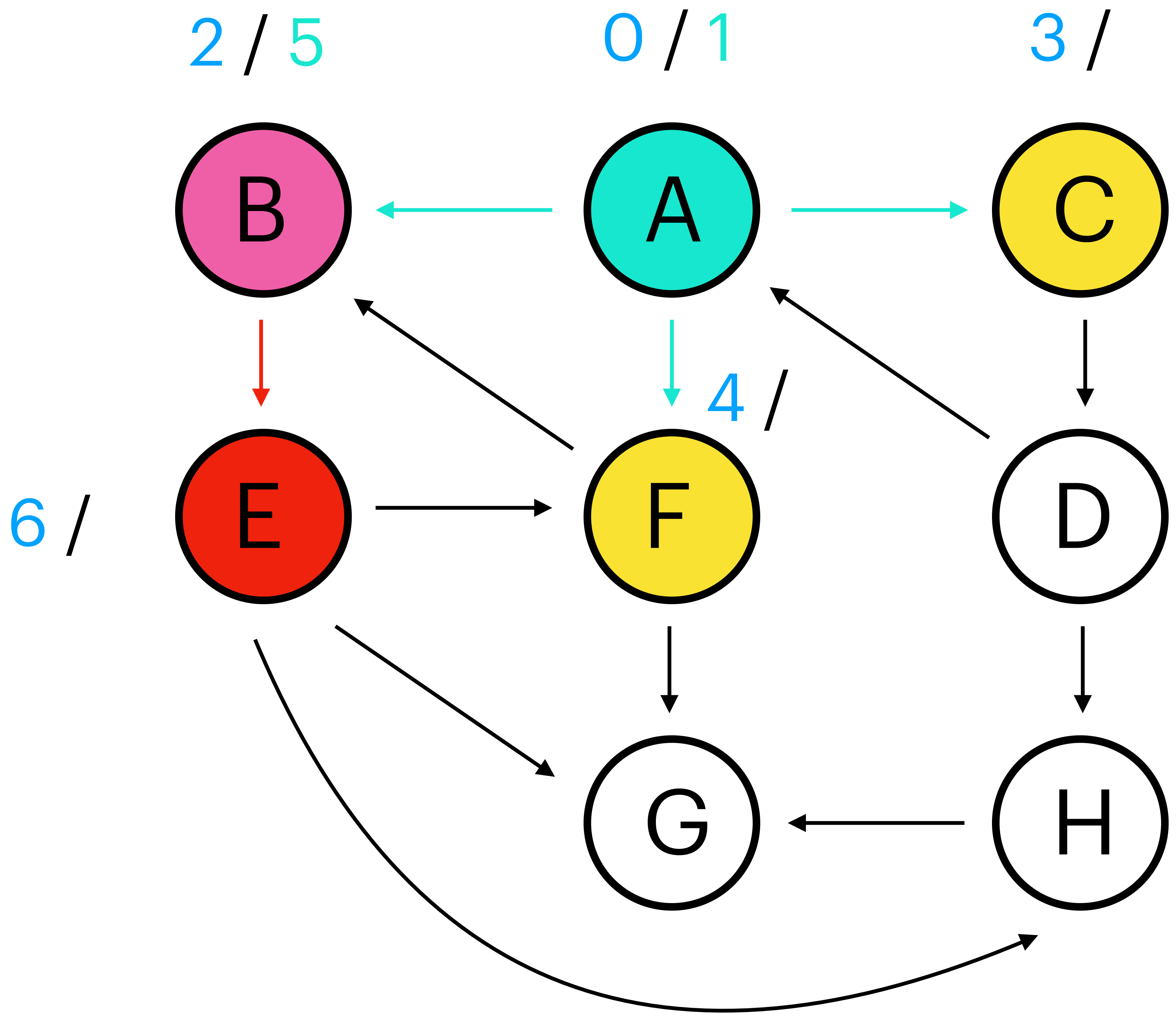


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : C - F - E

u = B



enter[] :

A	B	C	D	E	F	G	H
0	2	3		6	4		

leave[] :

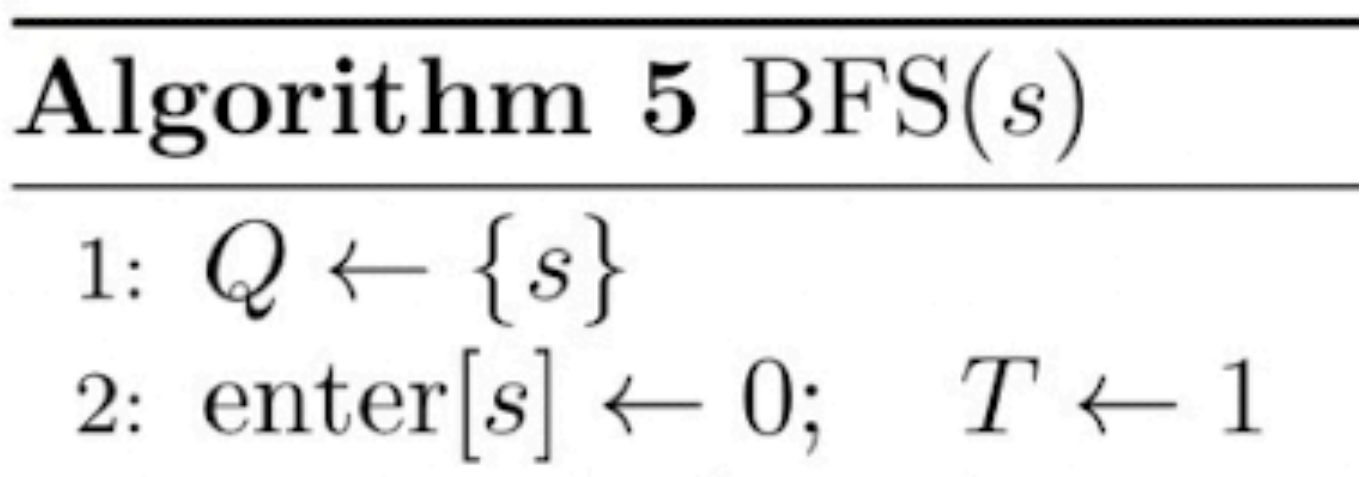
A	B	C	D	E	F	G	H
1	5						

distance[] :

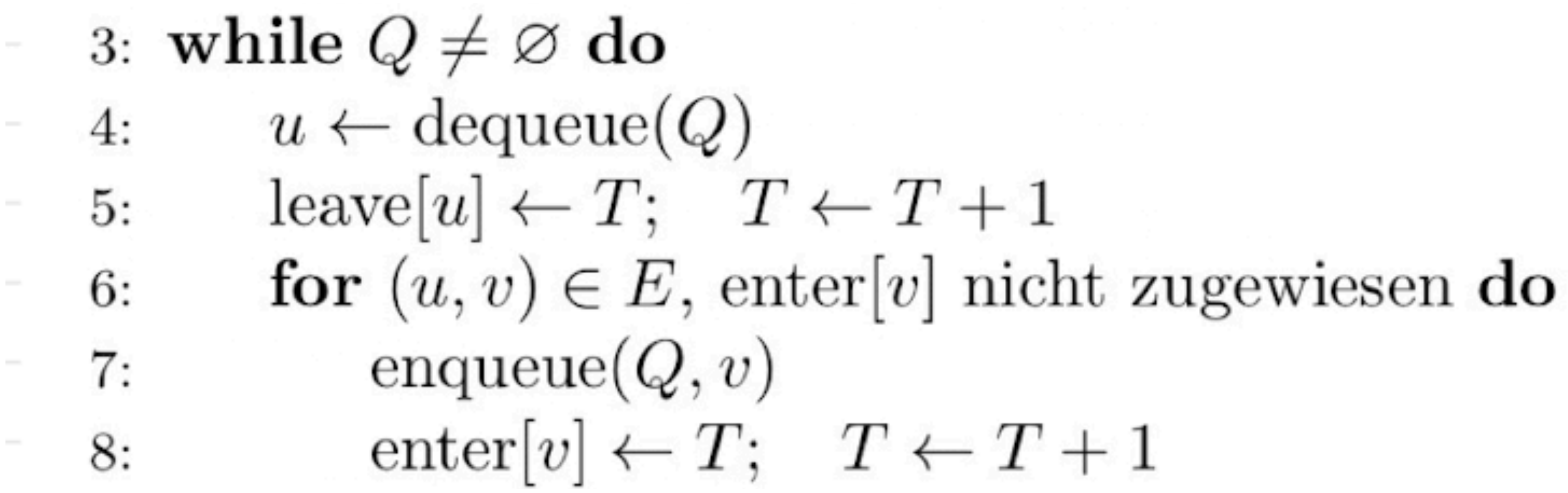
A	B	C	D	E	F	G	H
0	1	1		2	1		

Graph Searchs

BFS - Example

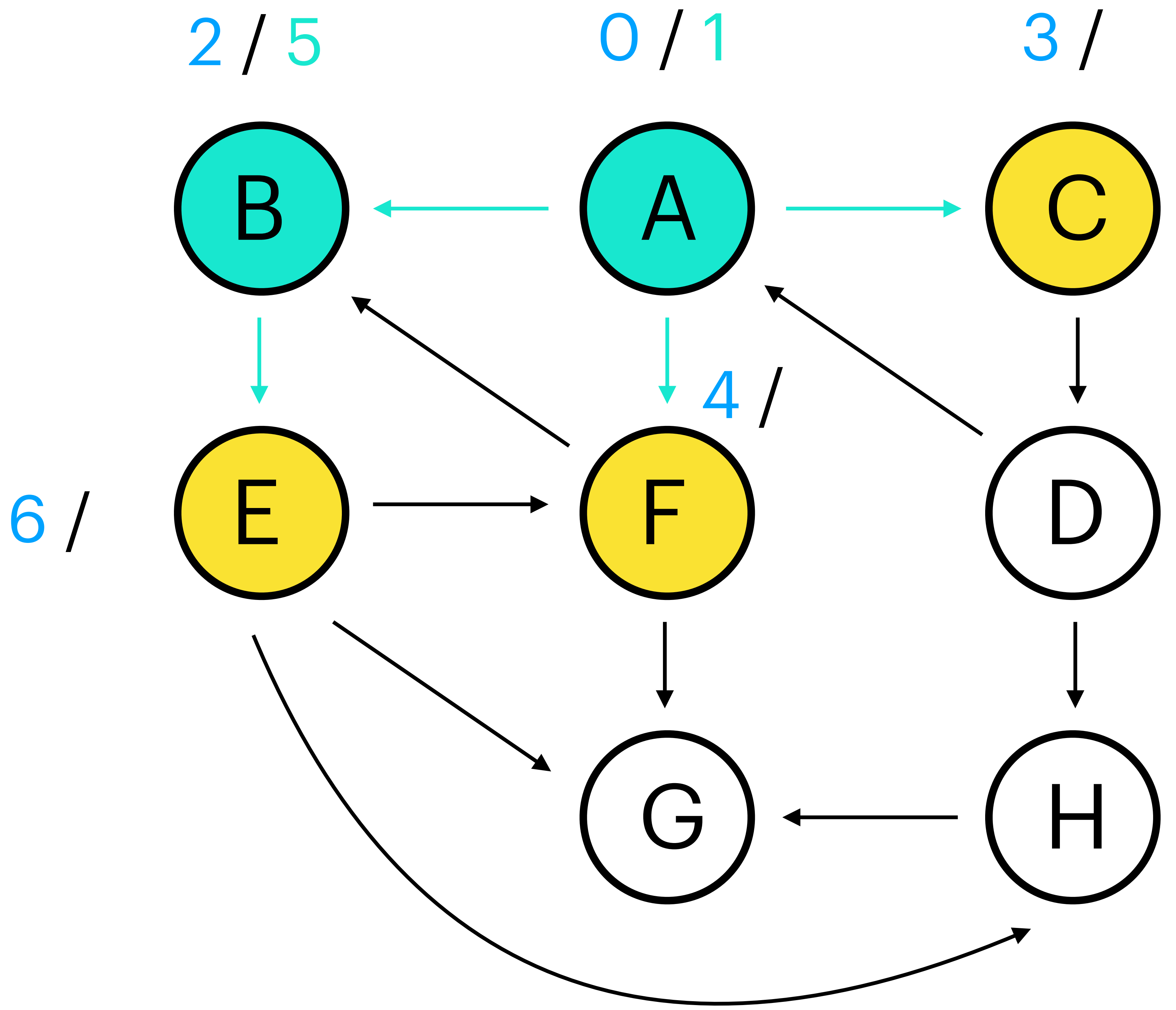


```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : C - F - E



enter[] :

A	B	C	D	E	F	G	H
0	2	3		6	4		

leave[] :

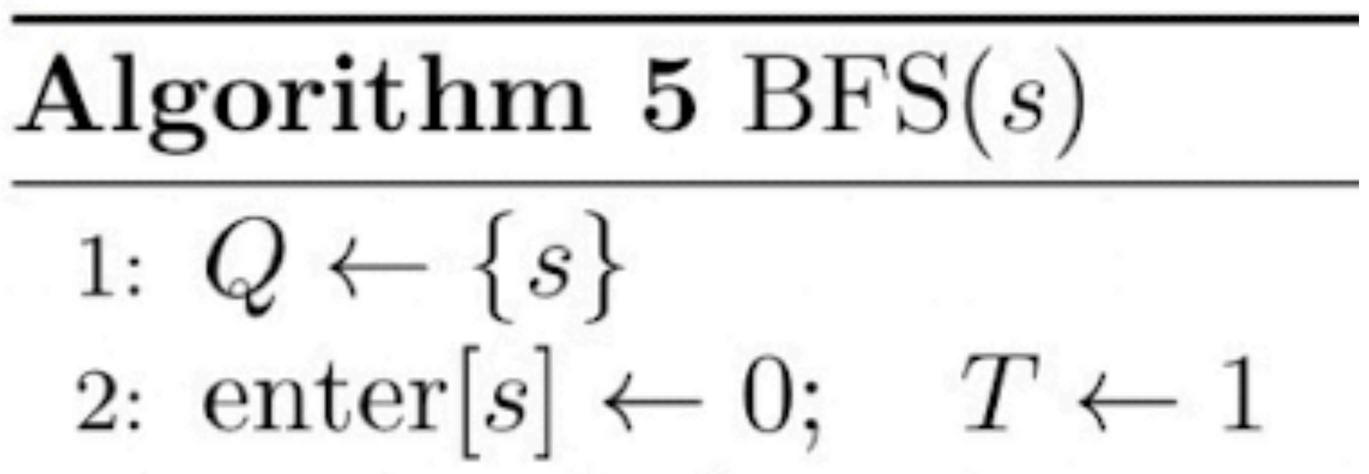
A	B	C	D	E	F	G	H
1	5						

distance[] :

A	B	C	D	E	F	G	H
0	1	1		2	1		

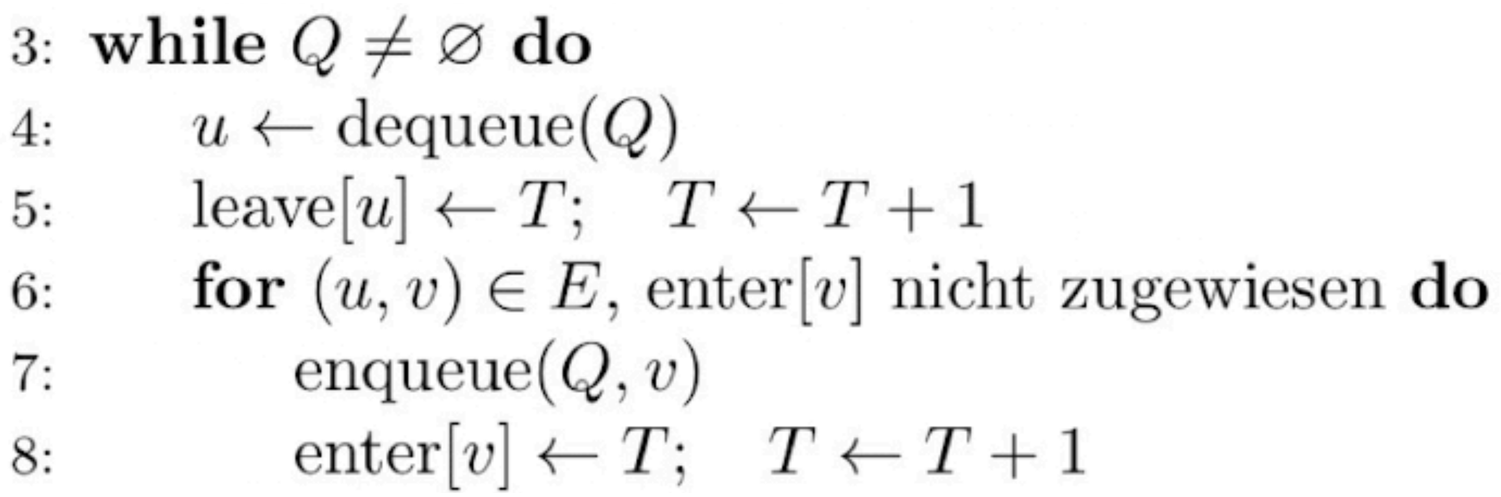
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

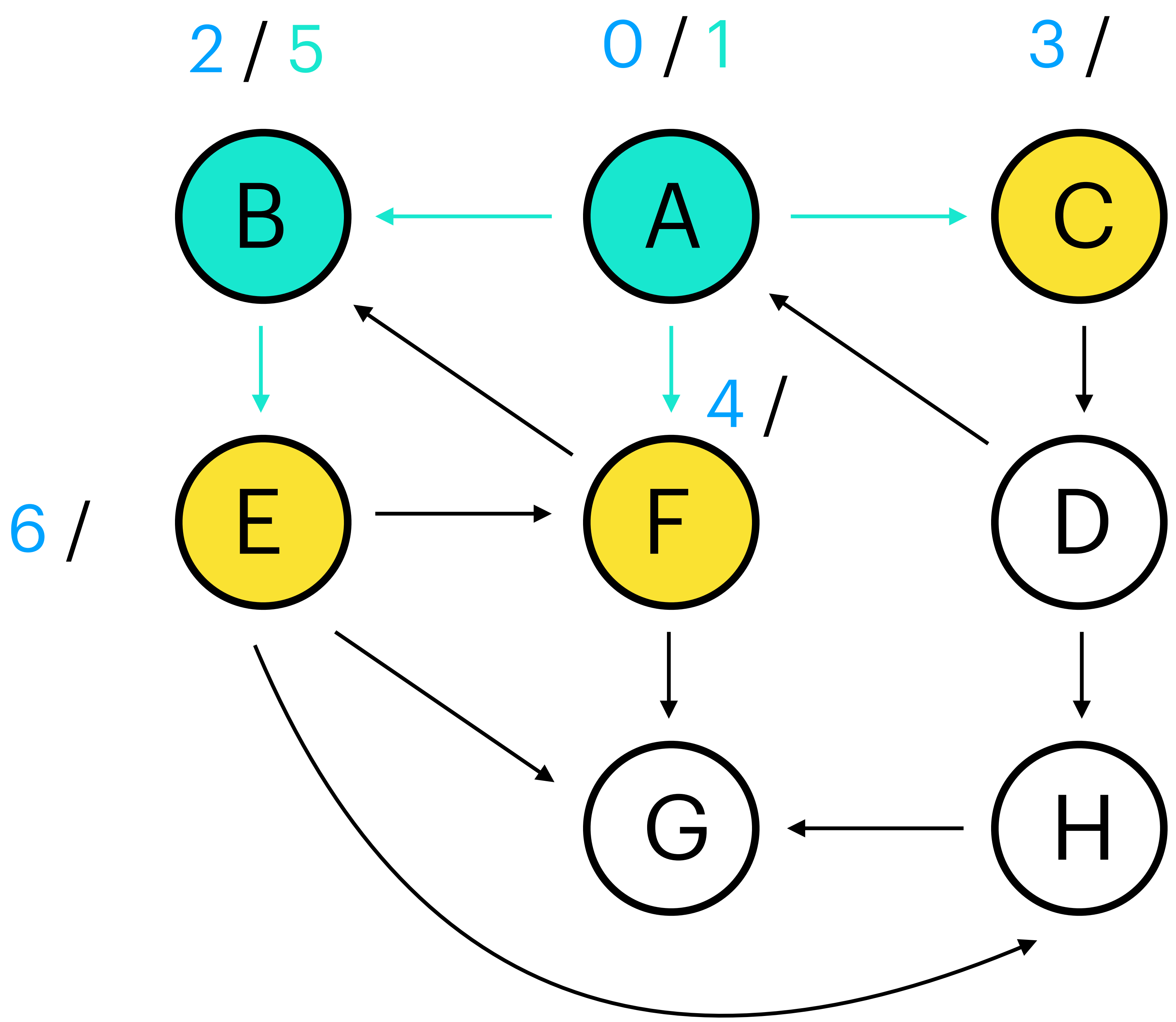
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : C - F - E



enter[] :

A	B	C	D	E	F	G	H
0	2	3		6	4		

leave[] :

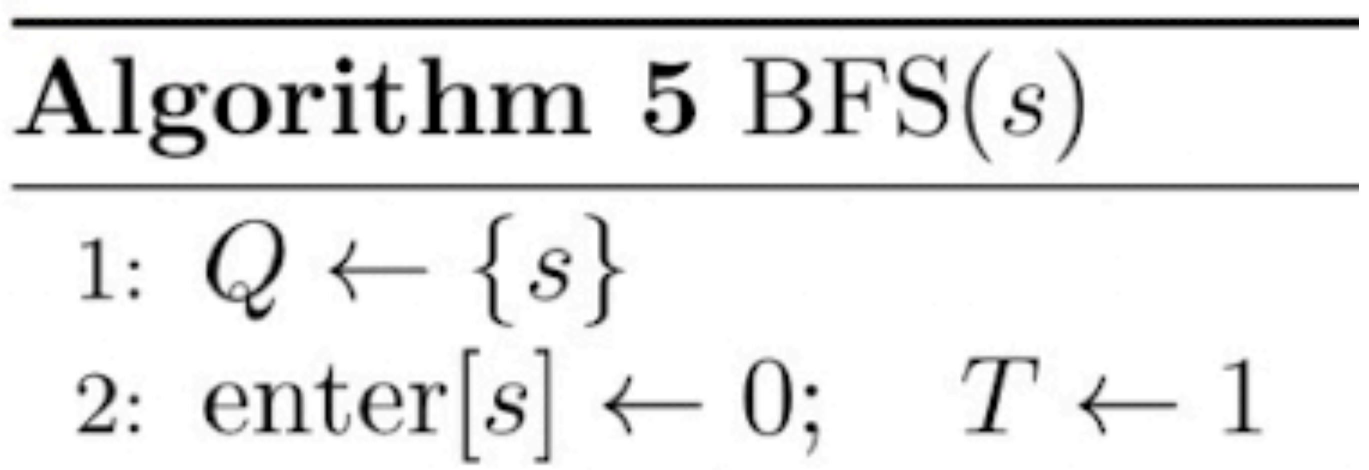
A	B	C	D	E	F	G	H
1	5						

distance[] :

A	B	C	D	E	F	G	H
0	1	1		2	1		

Graph Searchs

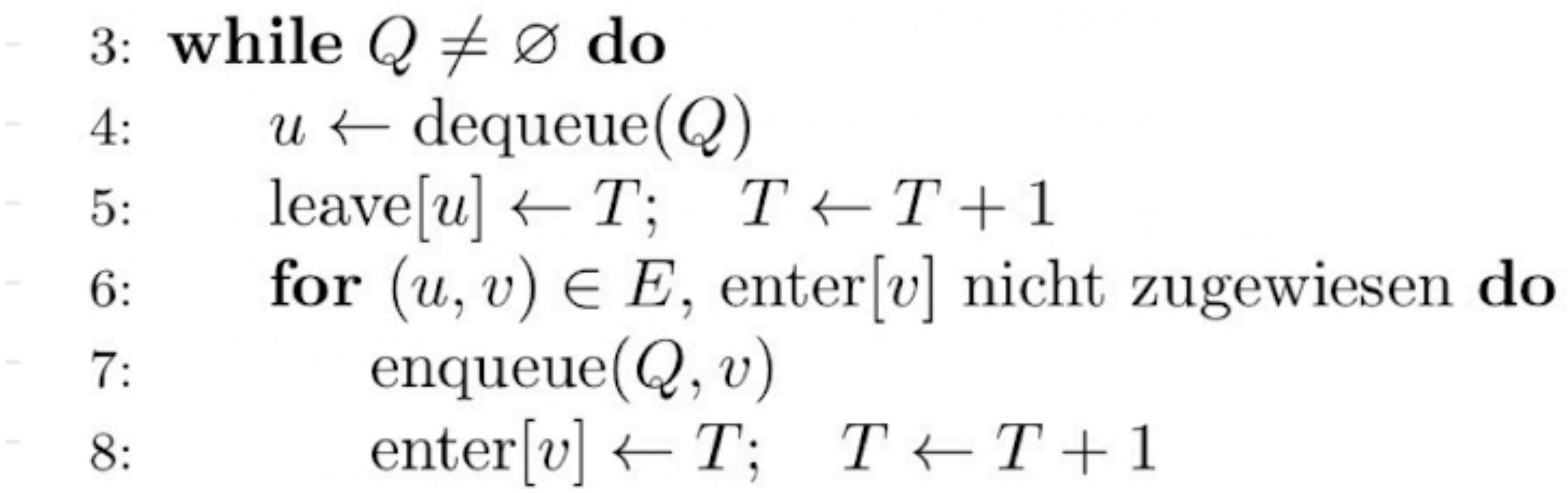
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

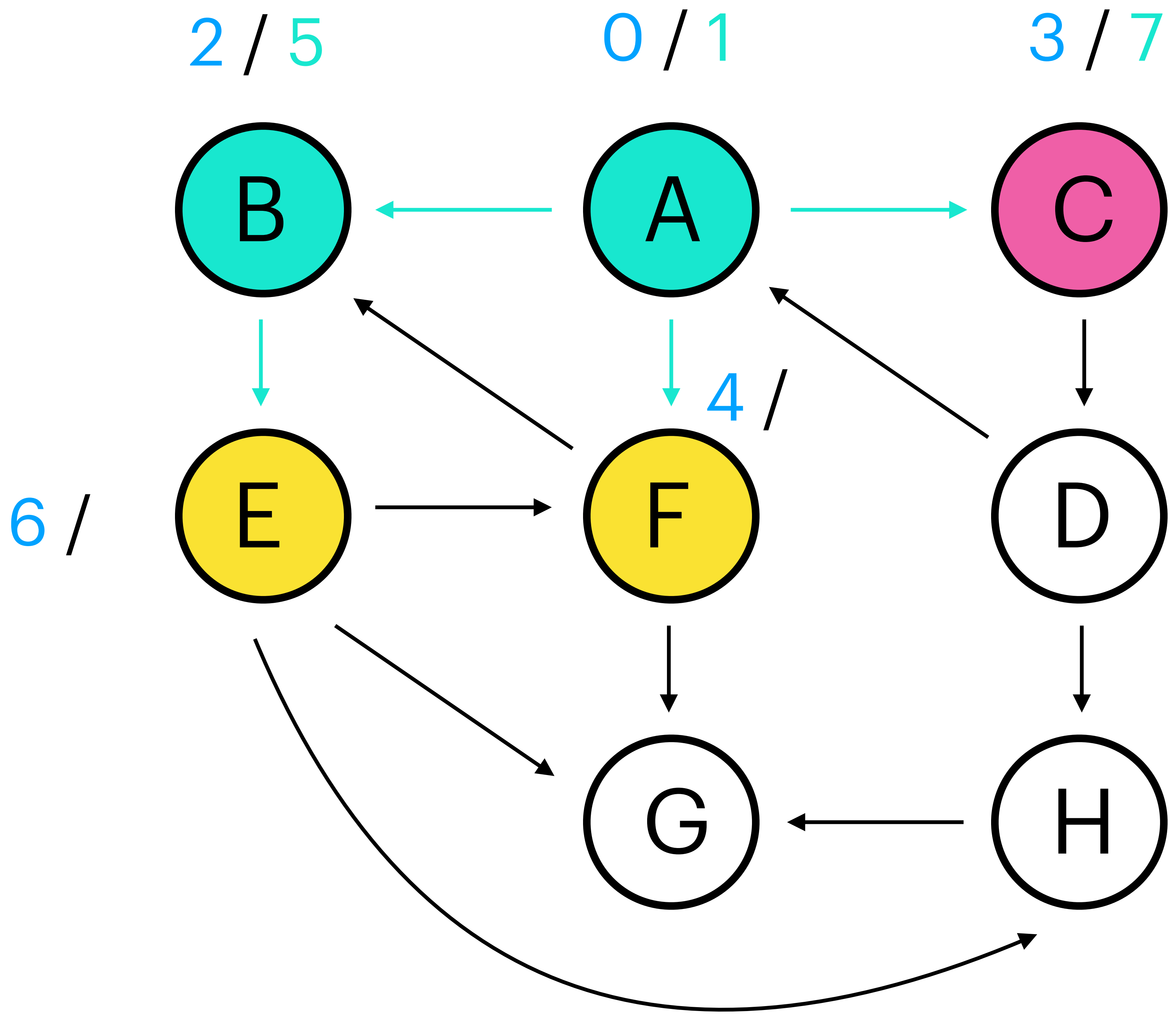


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : F - E

u = C



enter[] :

A	B	C	D	E	F	G	H
0	2	3		6	4		

leave[] :

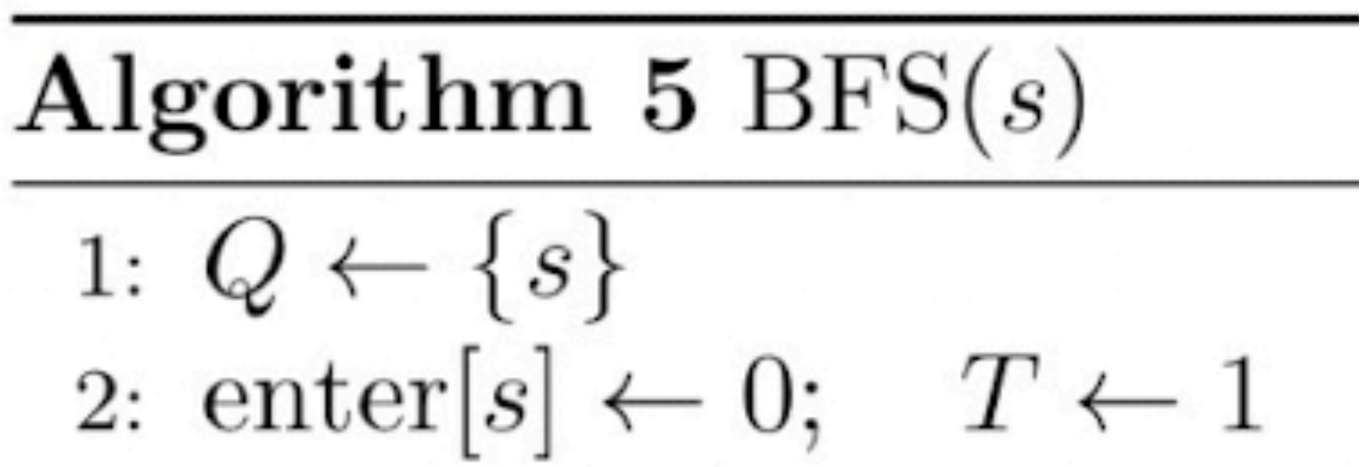
A	B	C	D	E	F	G	H
1	5	7					

distance[] :

A	B	C	D	E	F	G	H
0	1	1		2	1		

Graph Searchs

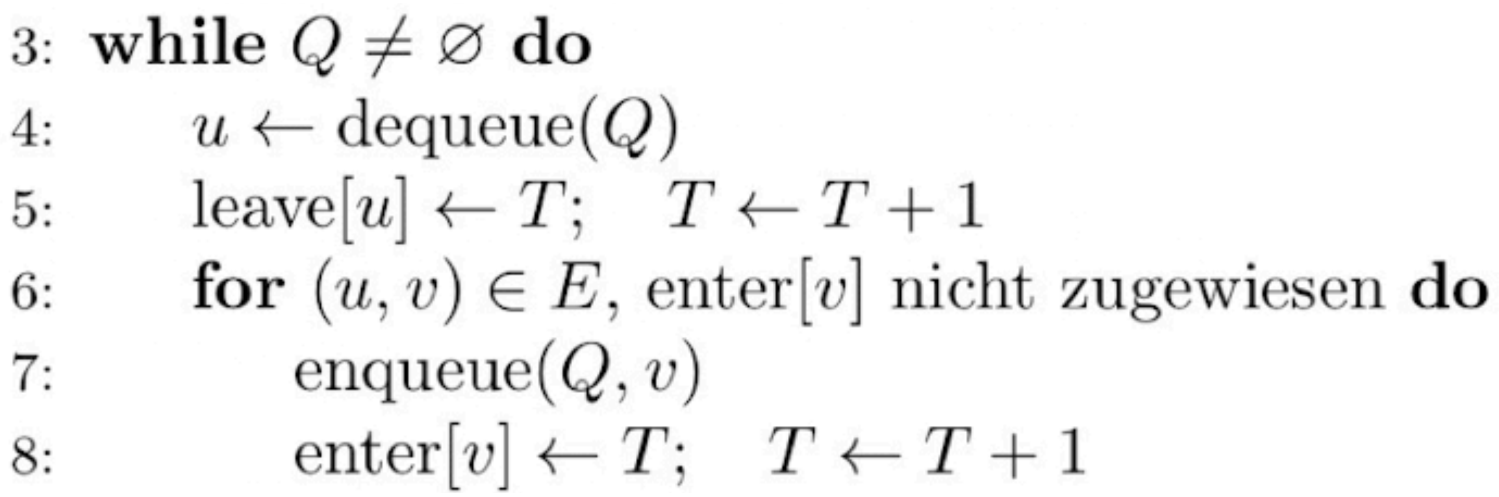
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : F - E

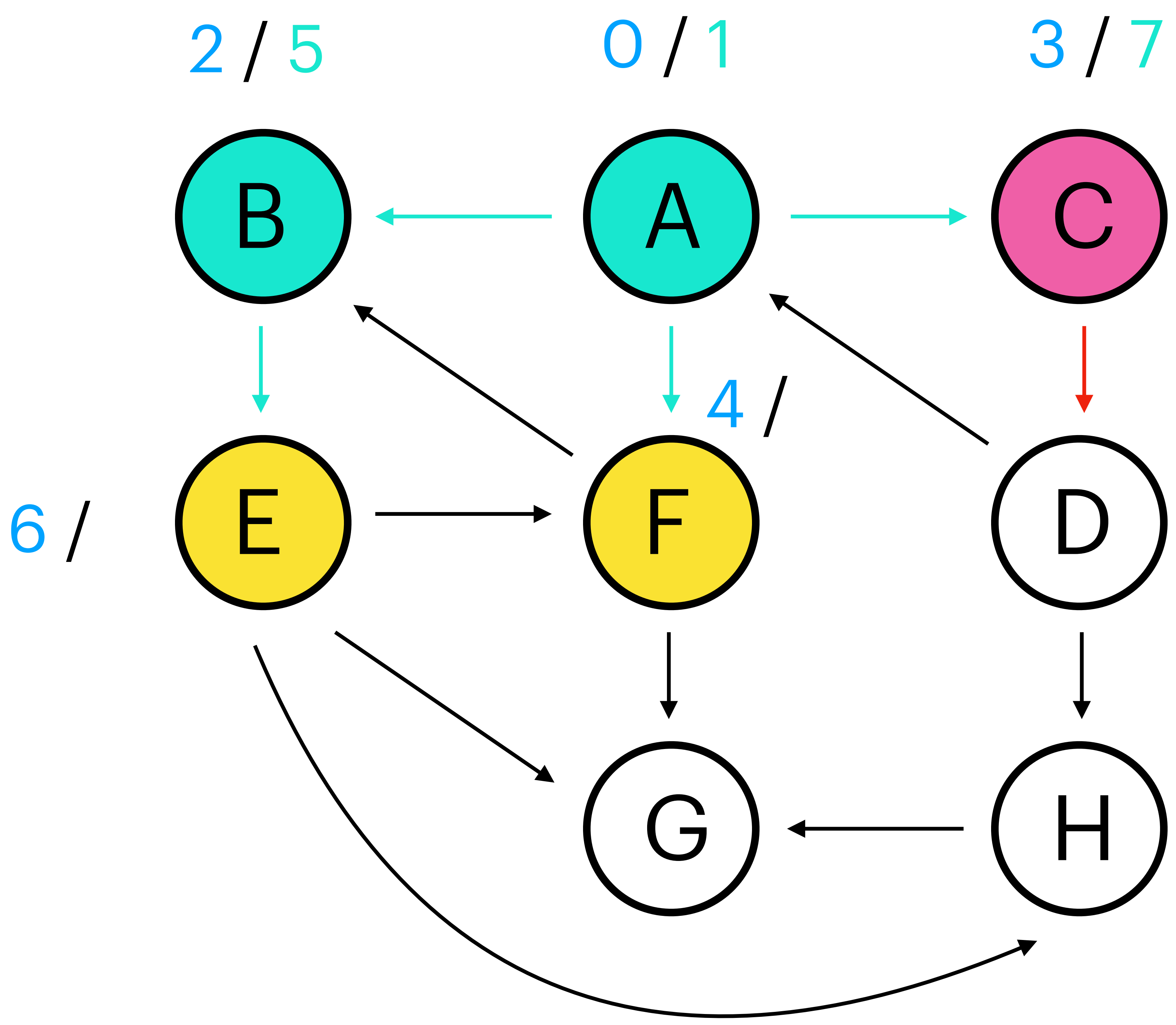
u = C

enter[] :

A	B	C	D	E	F	G	H
0	2	3		6	4		

leave[] :

A	B	C	D	E	F	G	H
1	5	7					

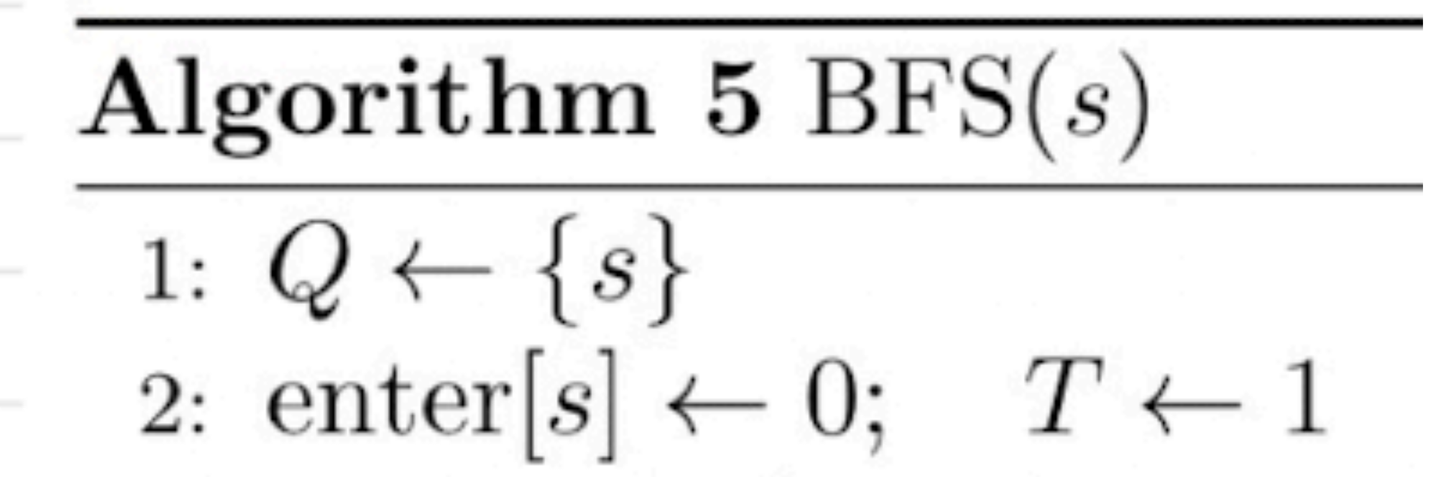


distance[] :

A	B	C	D	E	F	G	H
0	1	1		2	1		

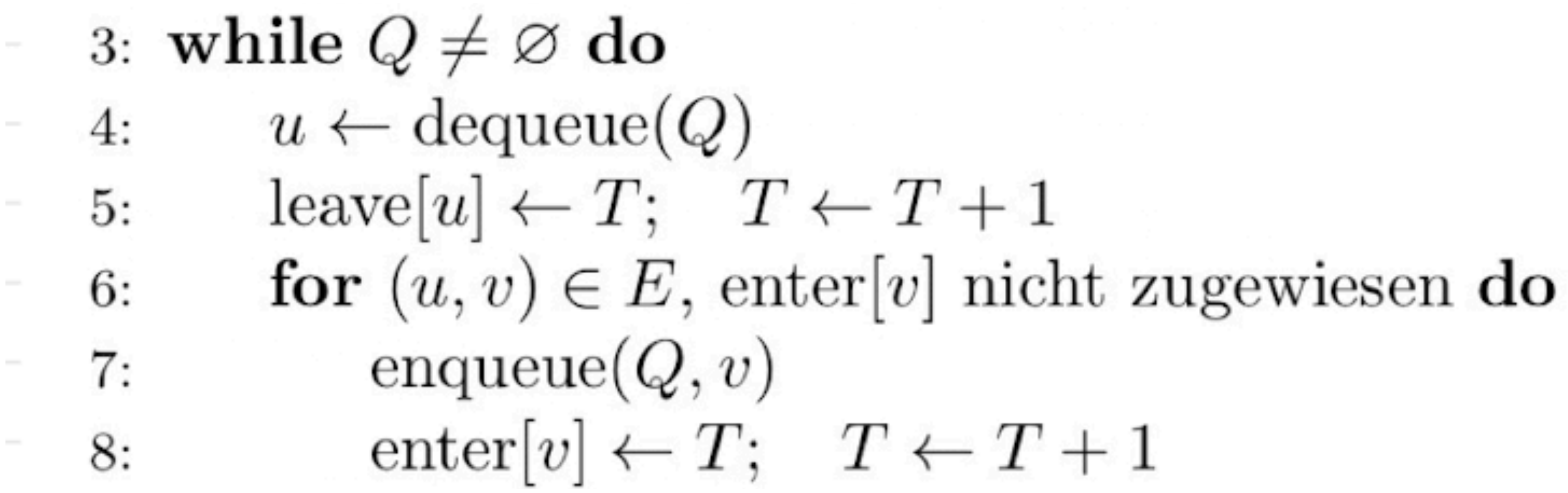
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

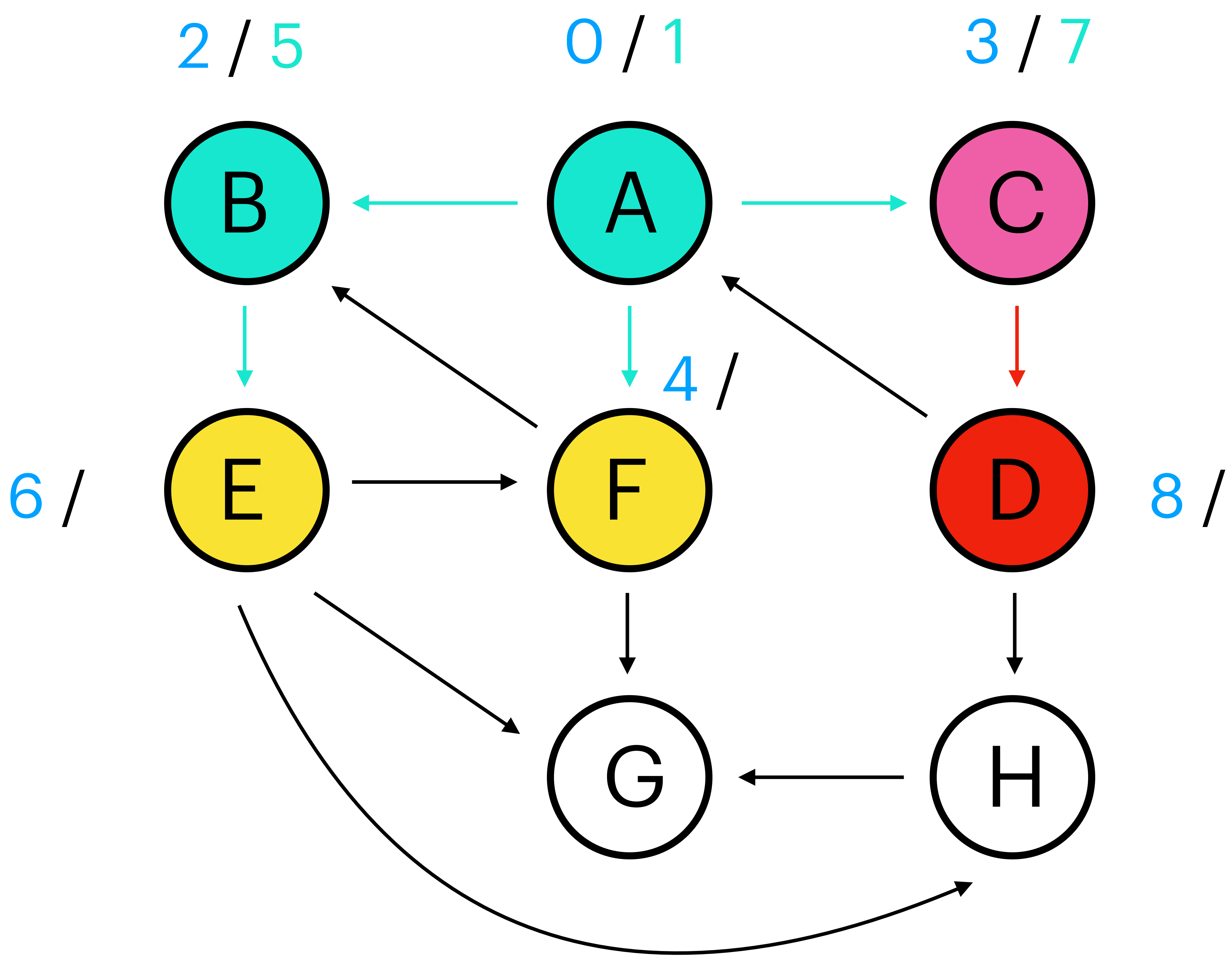


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : F - E - D

u = C



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4		

leave[] :

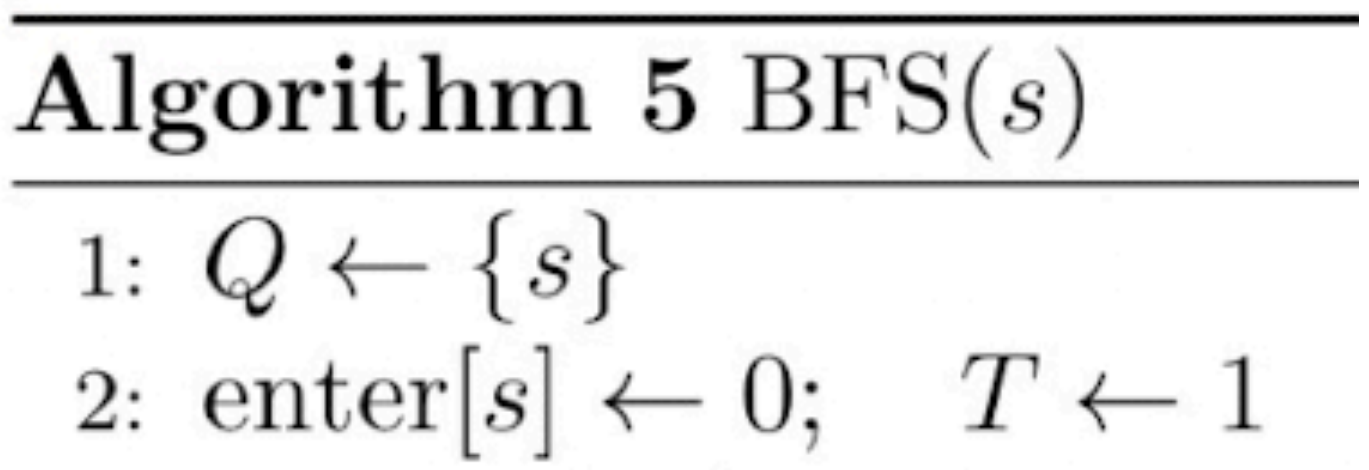
A	B	C	D	E	F	G	H
1	5	7					

distance[] :

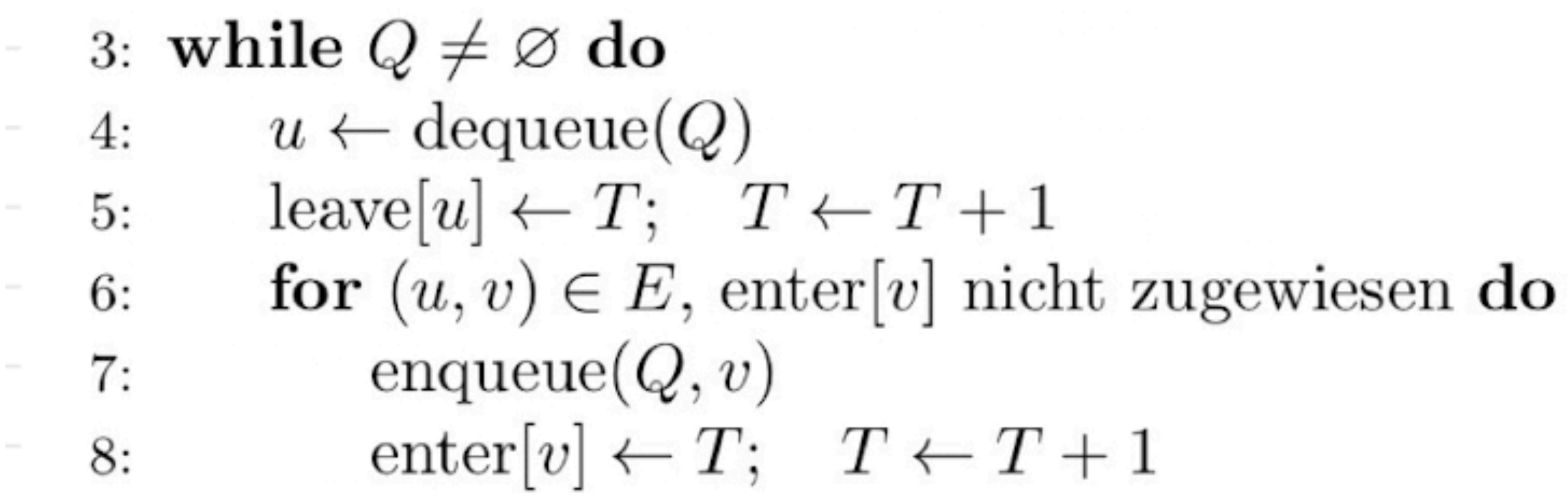
A	B	C	D	E	F	G	H
0	1	1	2	2	1		

Graph Searchs

BFS - Example

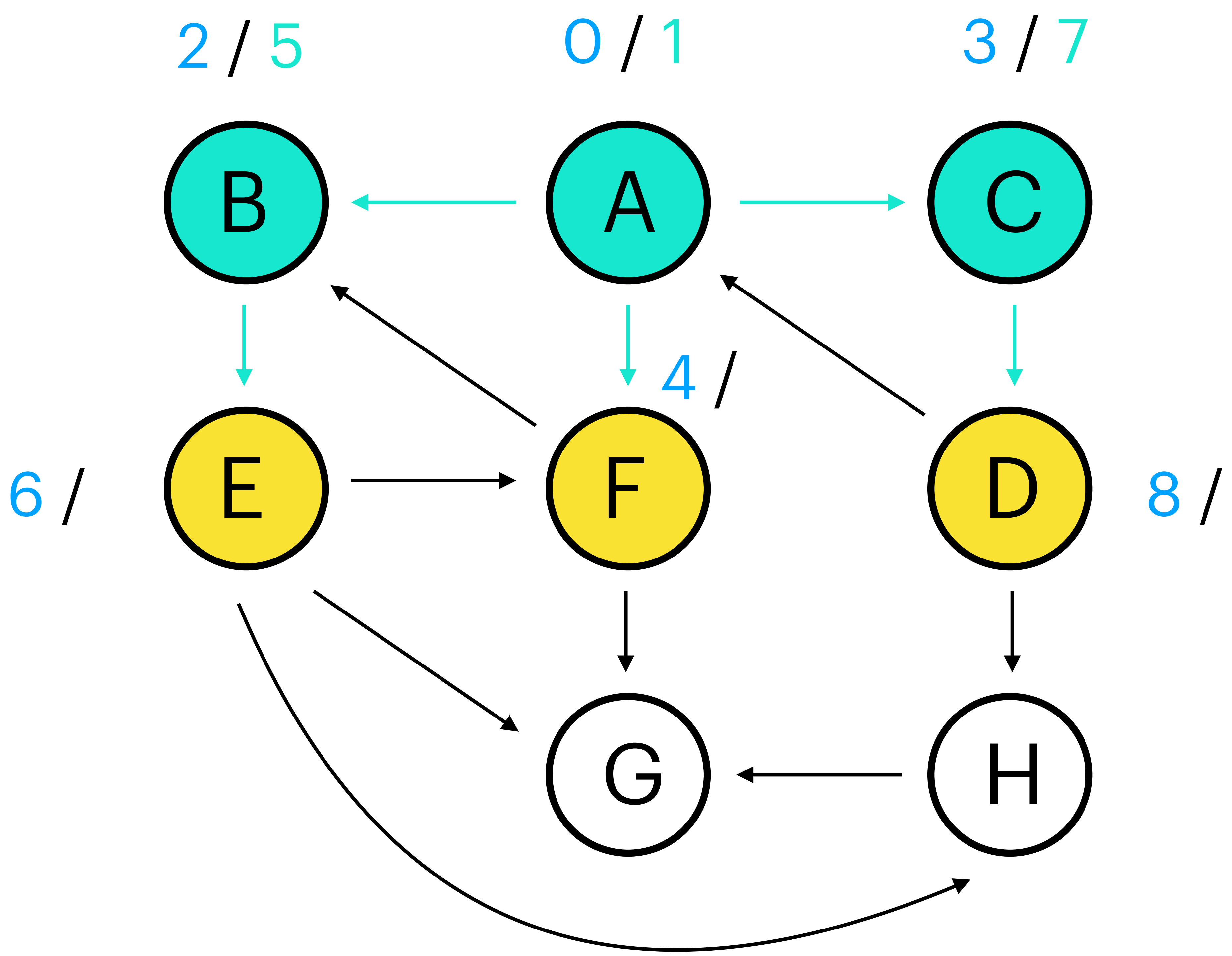


```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
    distance[s] = 0 ;
```



```
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : F - E - D



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4		

leave[] :

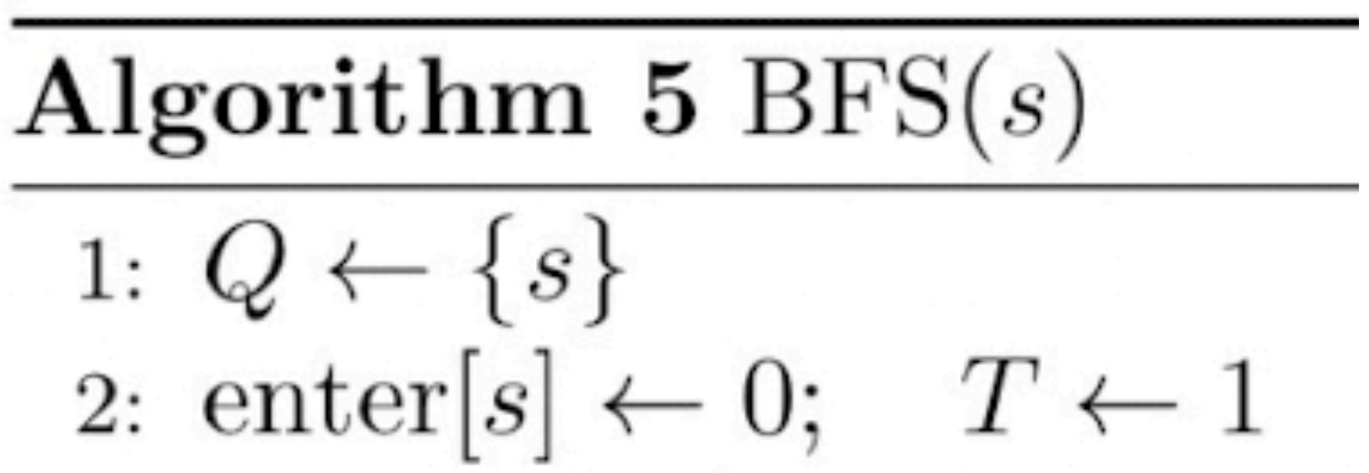
A	B	C	D	E	F	G	H
1	5	7					

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1		

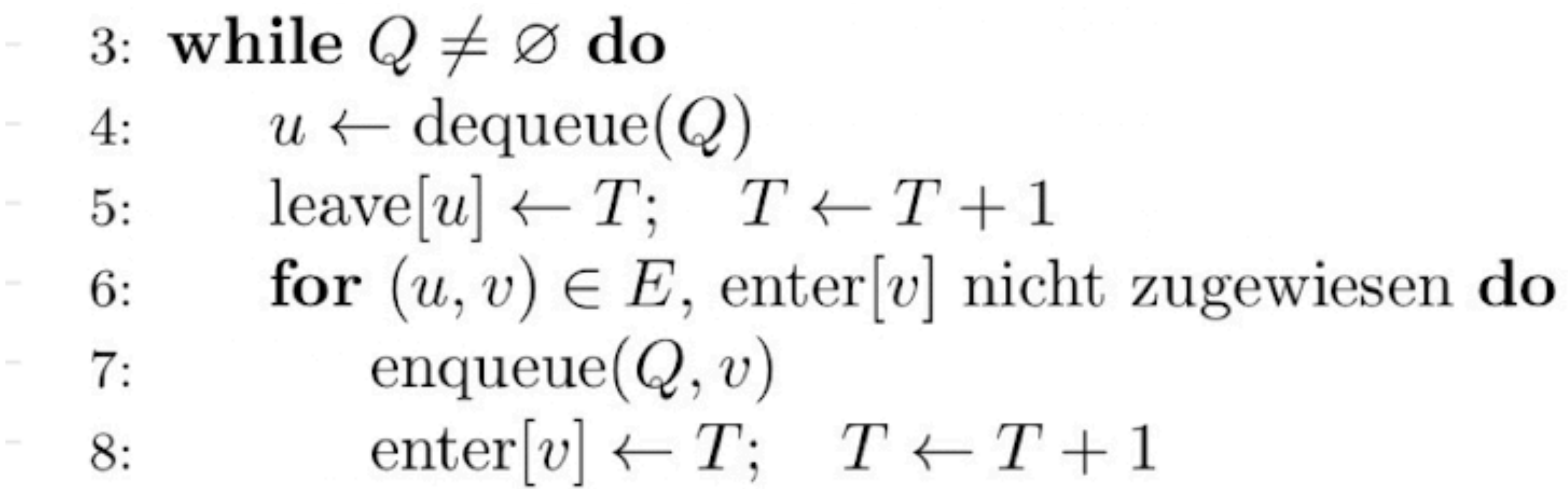
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
$\text{enter}[s] \leftarrow 0; \quad T \leftarrow 1$
```

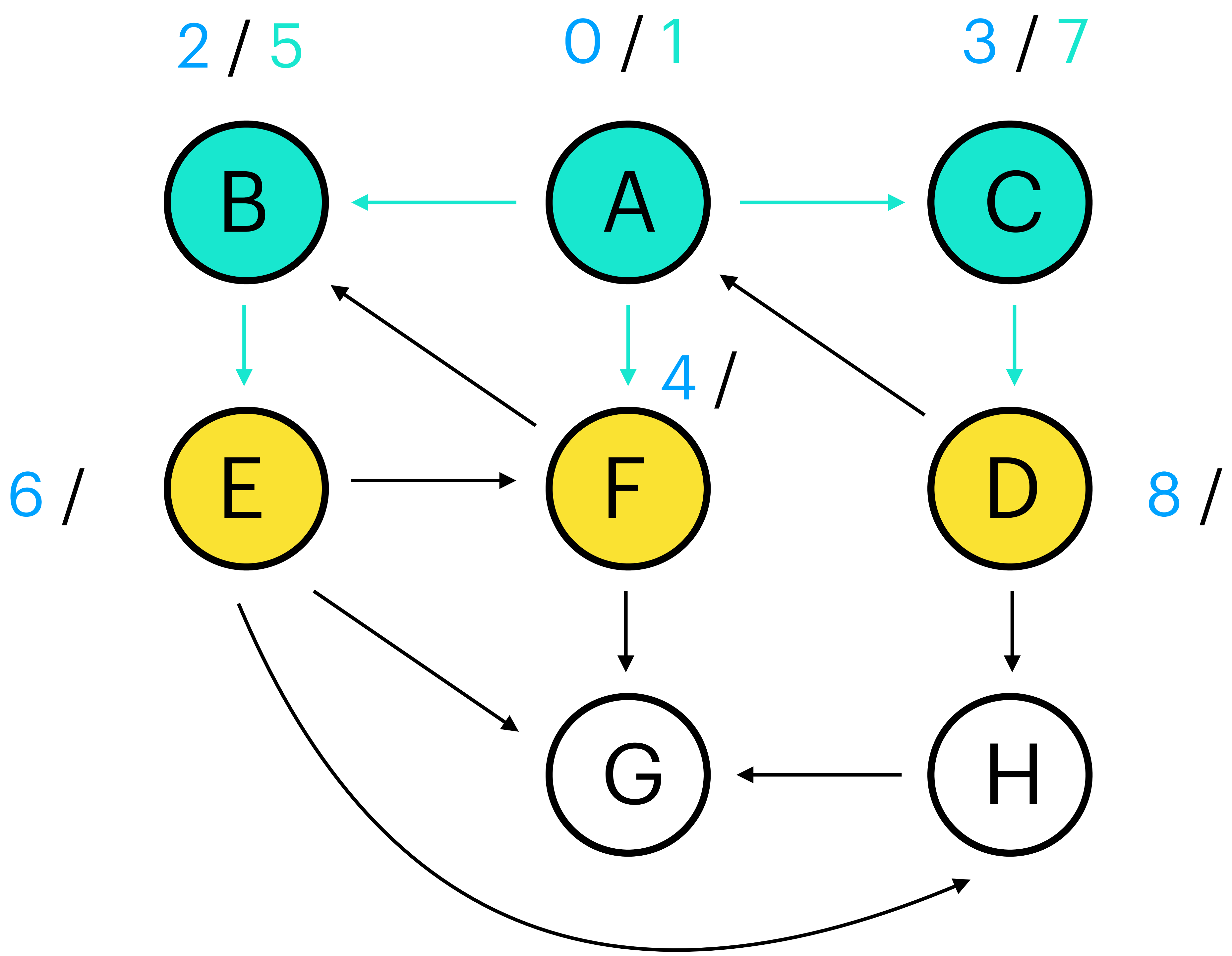
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow \text{dequeue}(Q)$
    $\text{leave}[u] \leftarrow T; \quad T \leftarrow T+1$
    for $(u,v) \in E$, $\text{enter}[v]$ nicht zugewiesen do
        $\text{enqueue}(Q,v)$
        $\text{enter}[v] \leftarrow T; \quad T \leftarrow T+1$
```

distance[v] <- distance[u] + 1 ;

Q : F - E - D



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4		

leave[] :

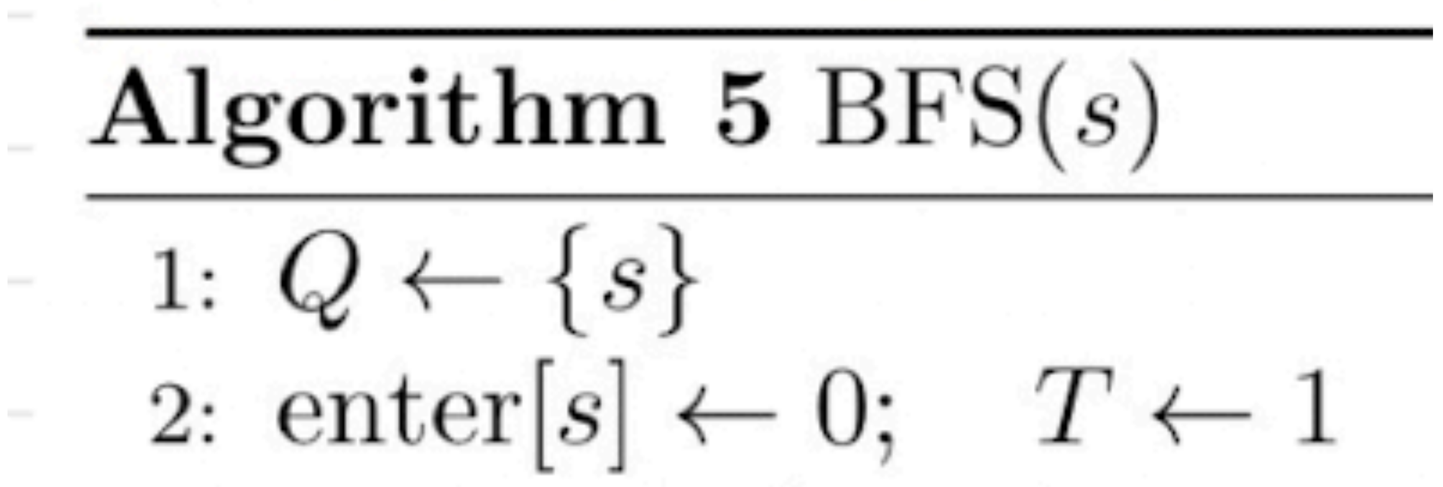
A	B	C	D	E	F	G	H
1	5	7					

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1		

Graph Searchs

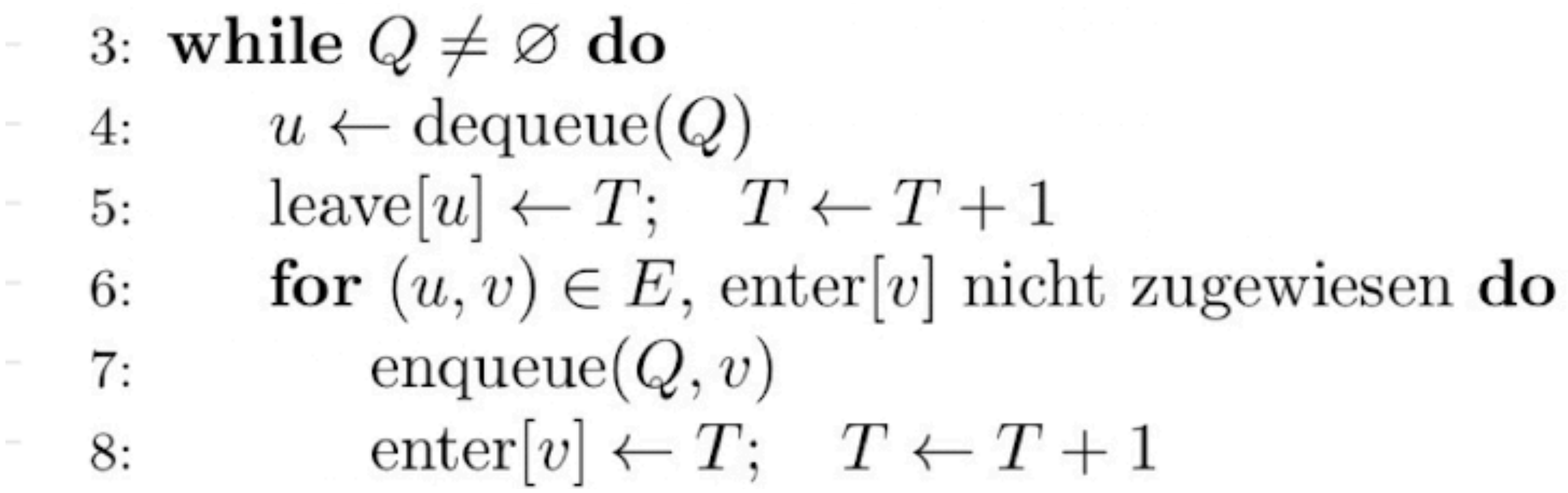
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

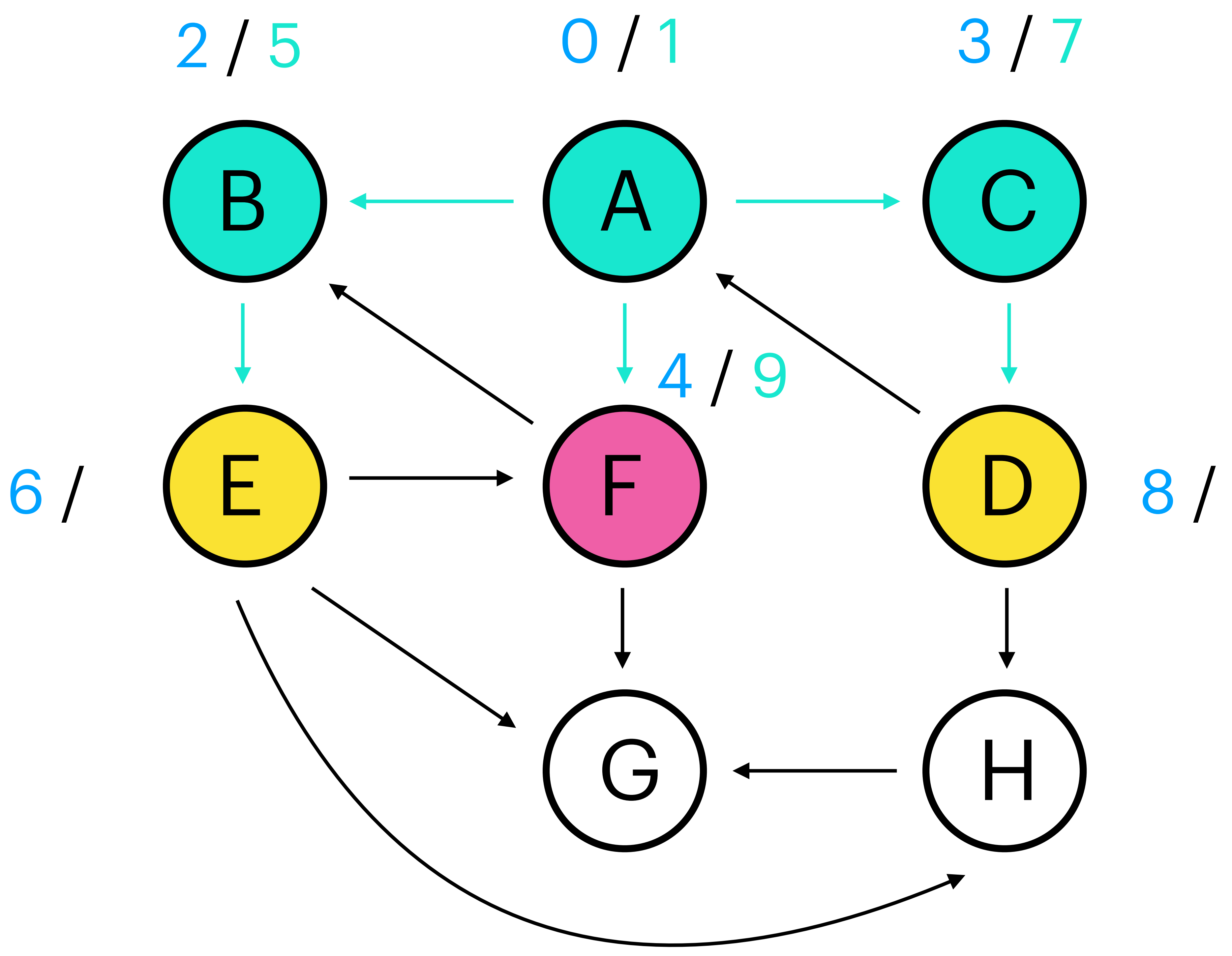


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : E - D

u = F



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4		

leave[] :

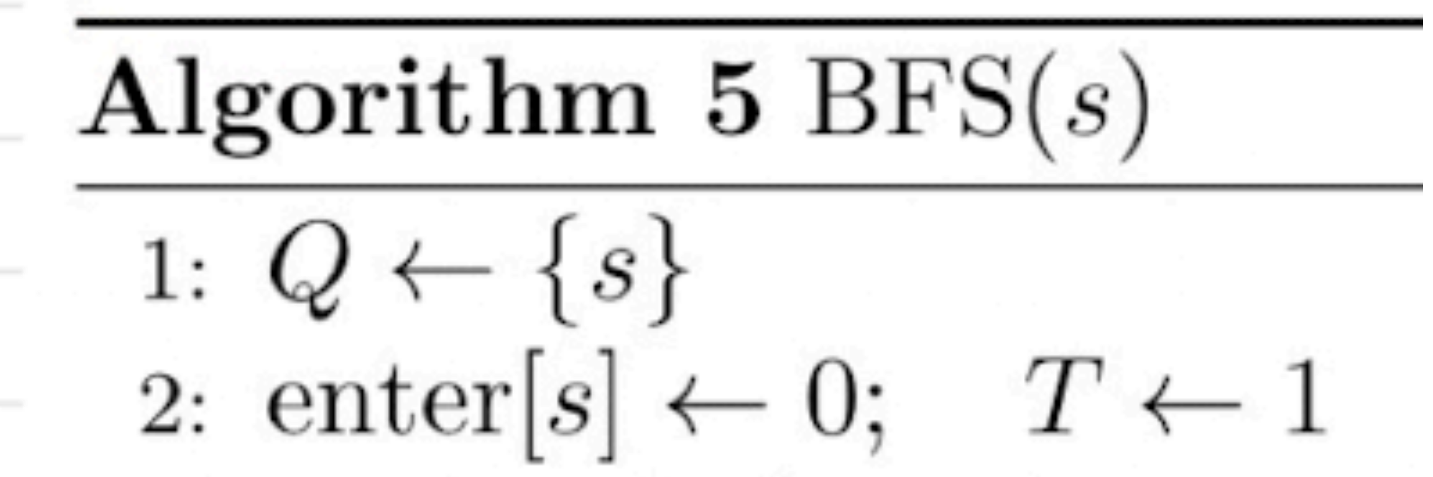
A	B	C	D	E	F	G	H
1	5	7			9		

distance[] :

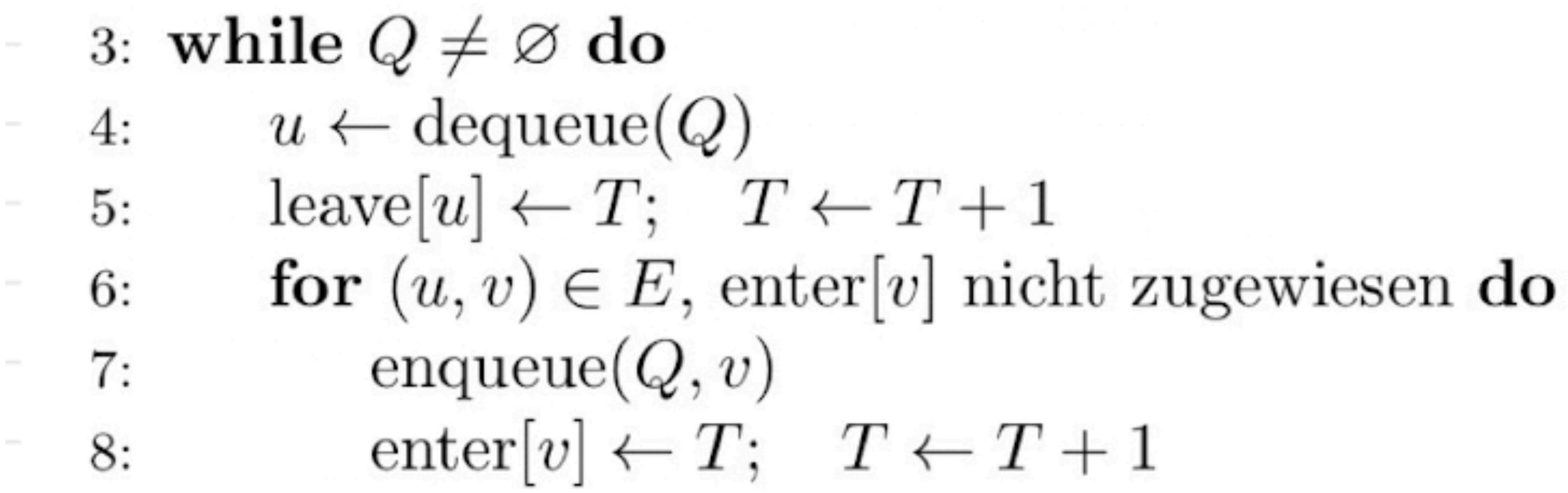
A	B	C	D	E	F	G	H
0	1	1	2	2	1		

Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
  $Q \leftarrow \{s\}$
  enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
  distance[s] = 0 ;
```



```
  while $Q \neq \varnothing$ do
      $u \leftarrow$ dequeue($Q$)
      leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
      for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
          enqueue($Q, v$)
          enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
          distance[v] <- distance[u] + 1 ;
```

Q : E - D

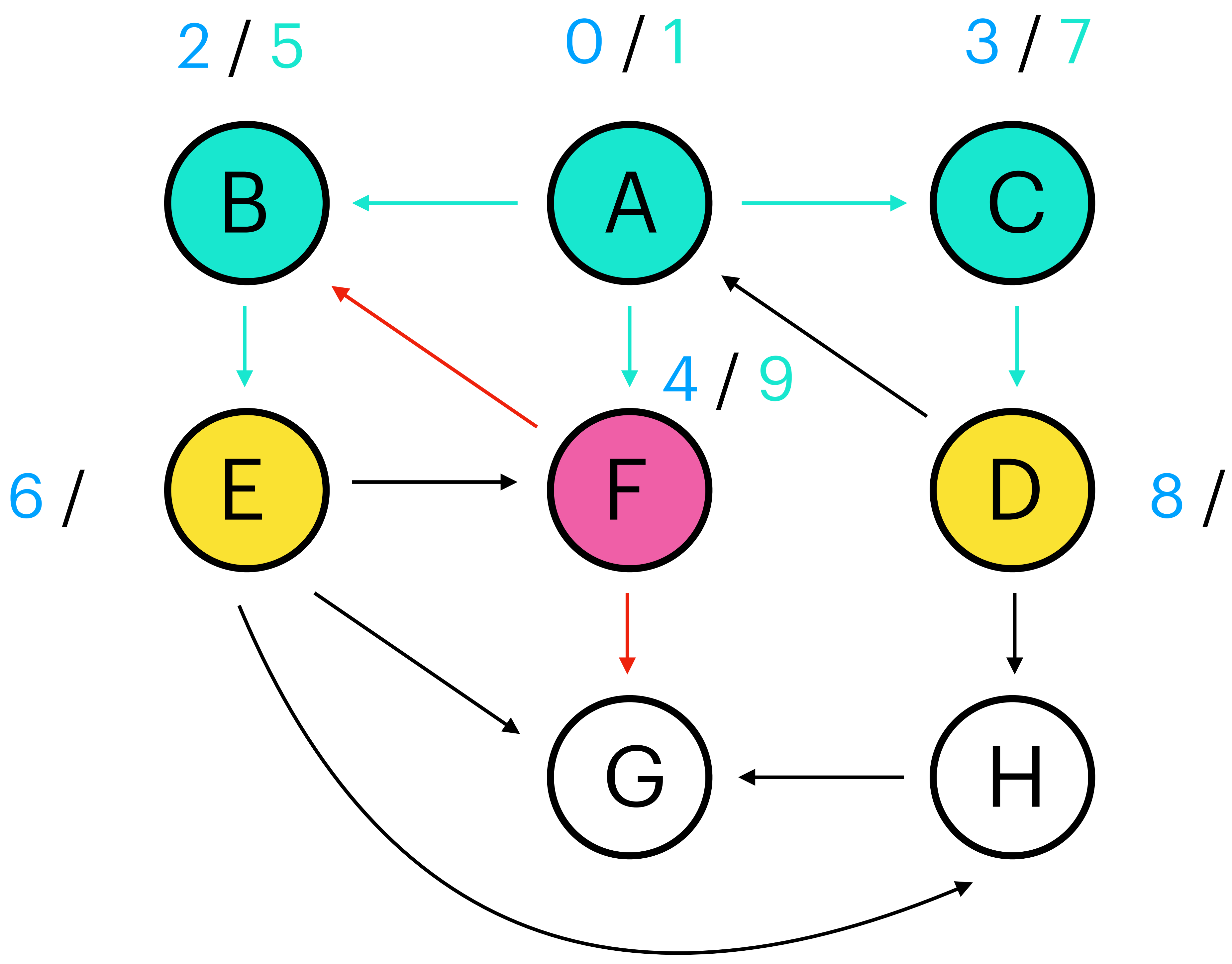
u = F

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4		

leave[] :

A	B	C	D	E	F	G	H
1	5	7			9		

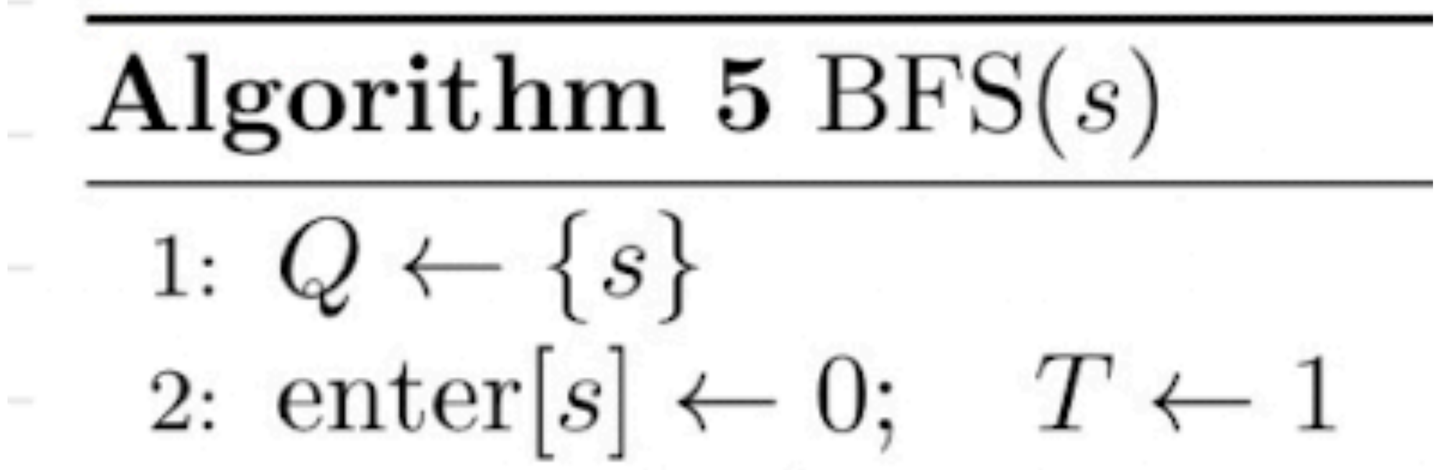


distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1		

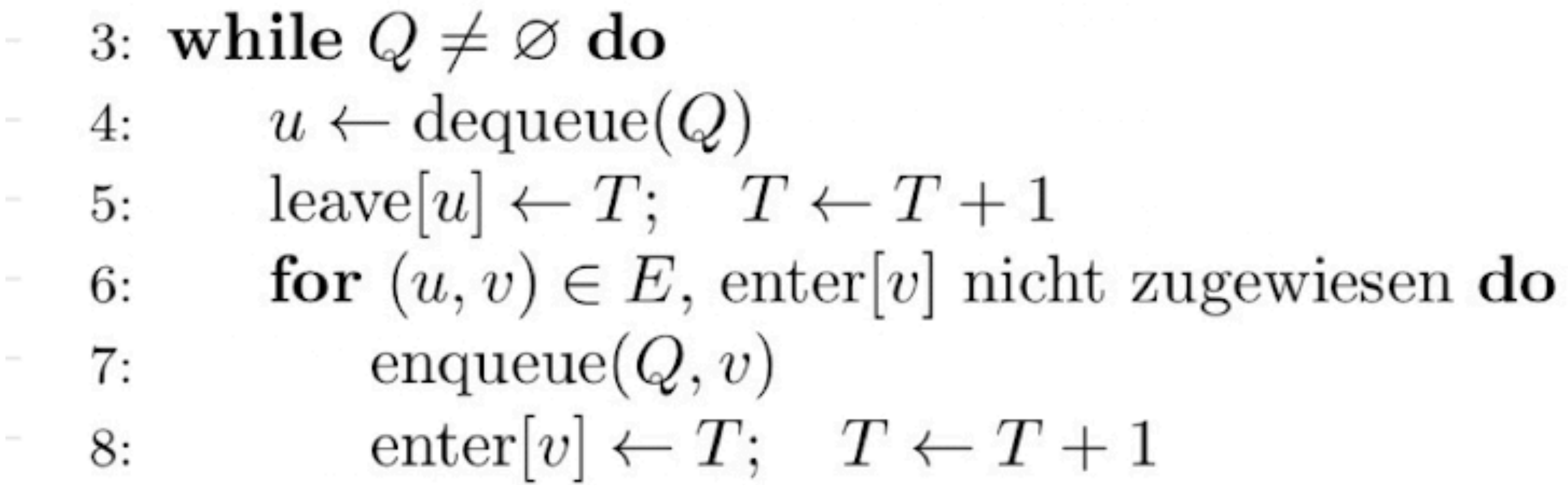
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

distance[s] = 0 ;

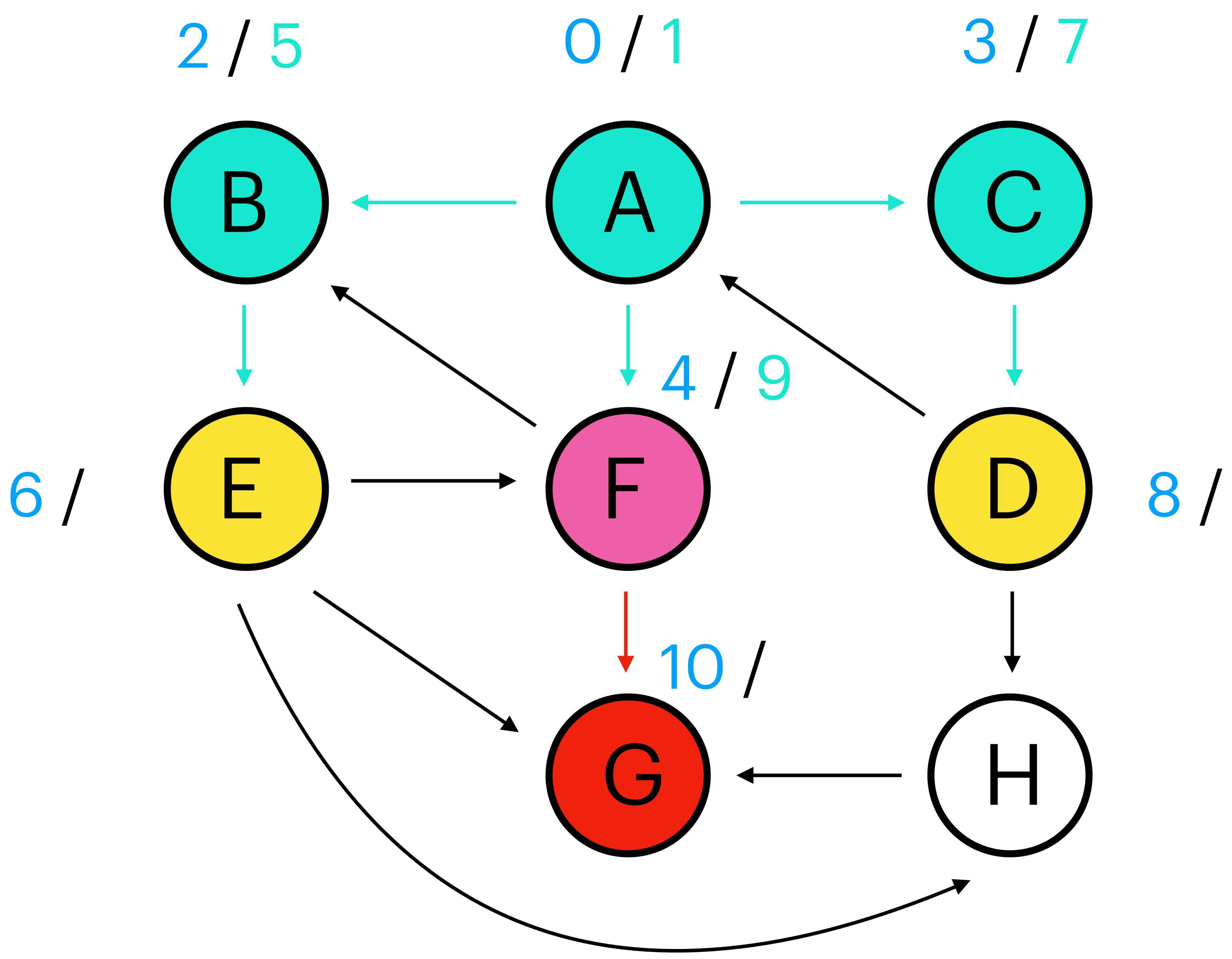


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : E - D - G

u = F



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	

leave[] :

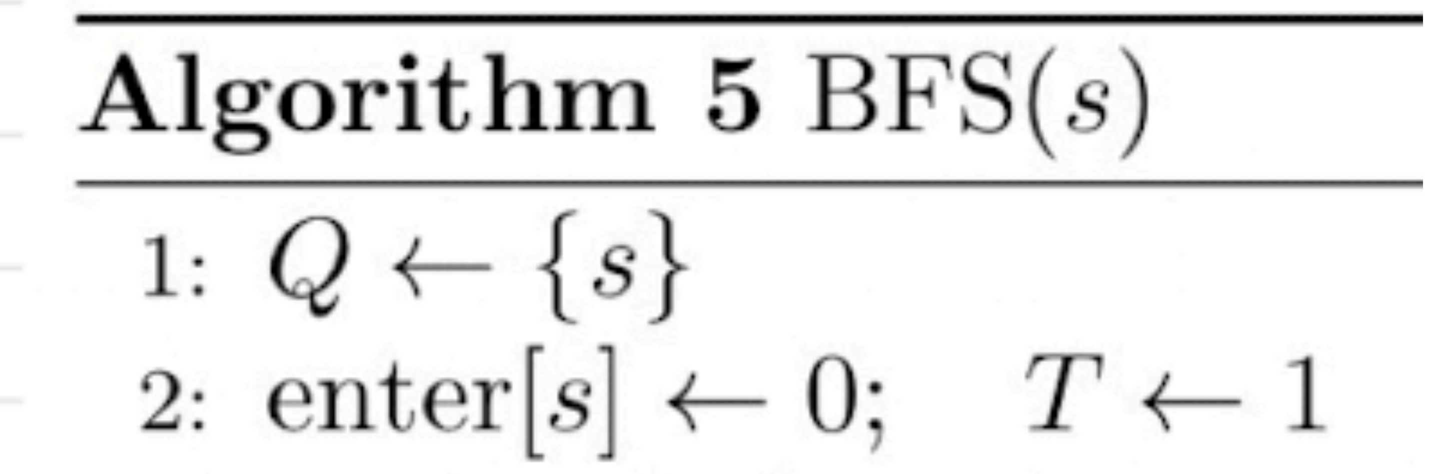
A	B	C	D	E	F	G	H
1	5	7			9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	

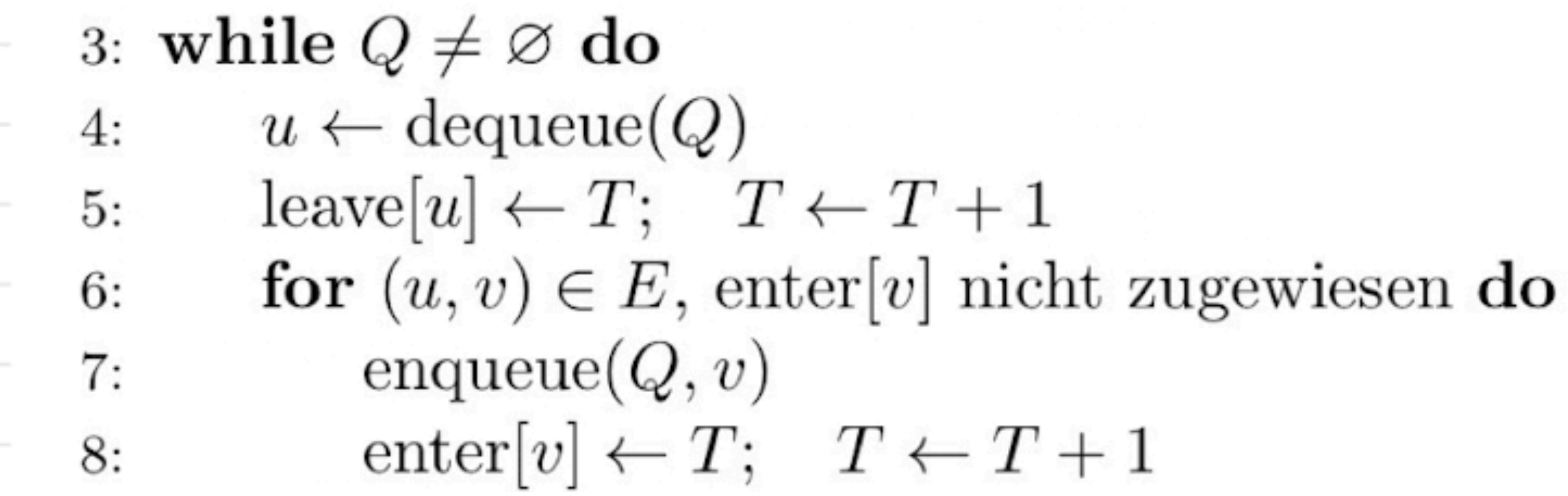
Graph Searchs

BFS - Example



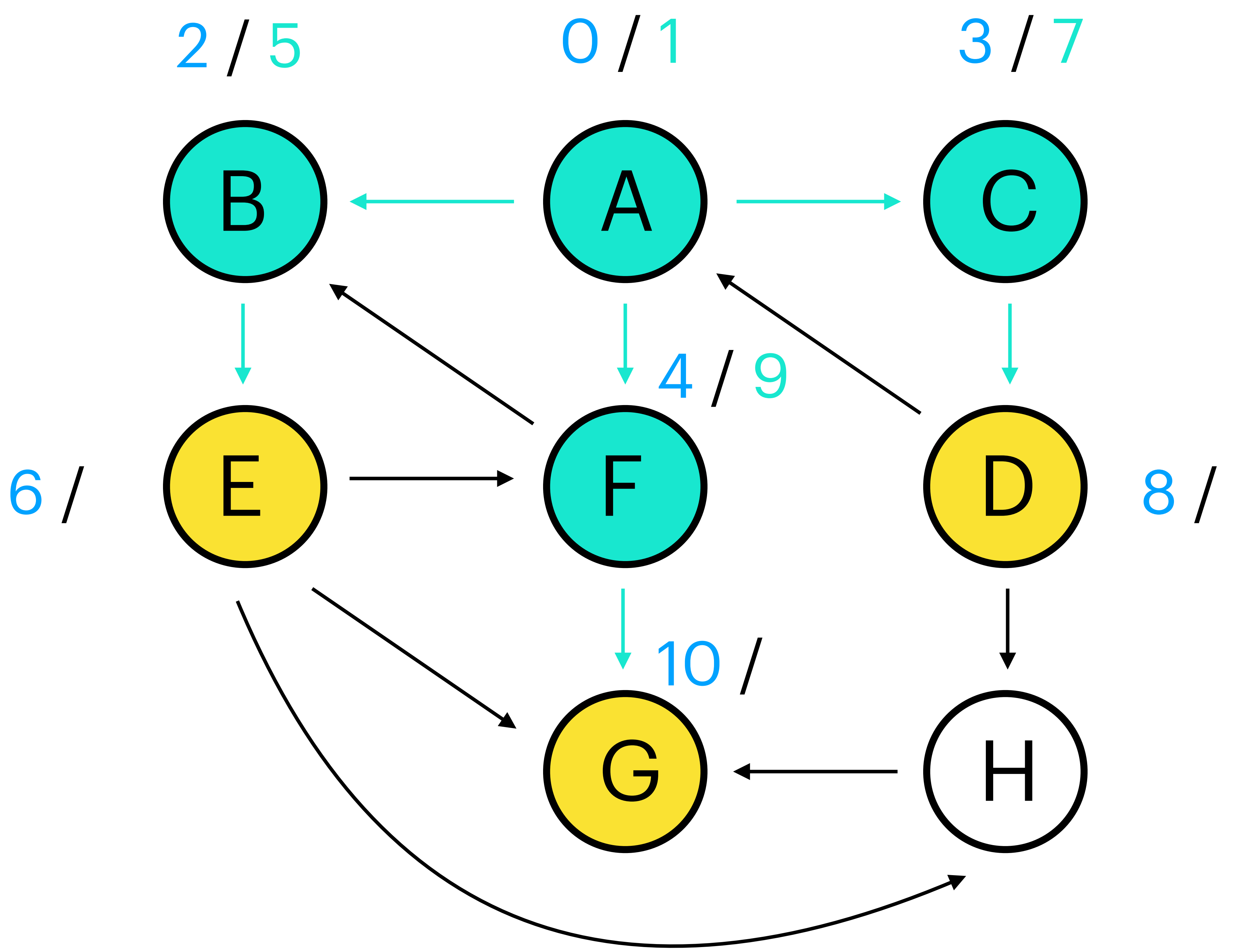
Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue$(Q)$
    leave$[u] \leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter$[v]$ nicht zugewiesen do
        enqueue$(Q, v)$
        enter$[v] \leftarrow T$;    $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : E - D - G



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	

leave[] :

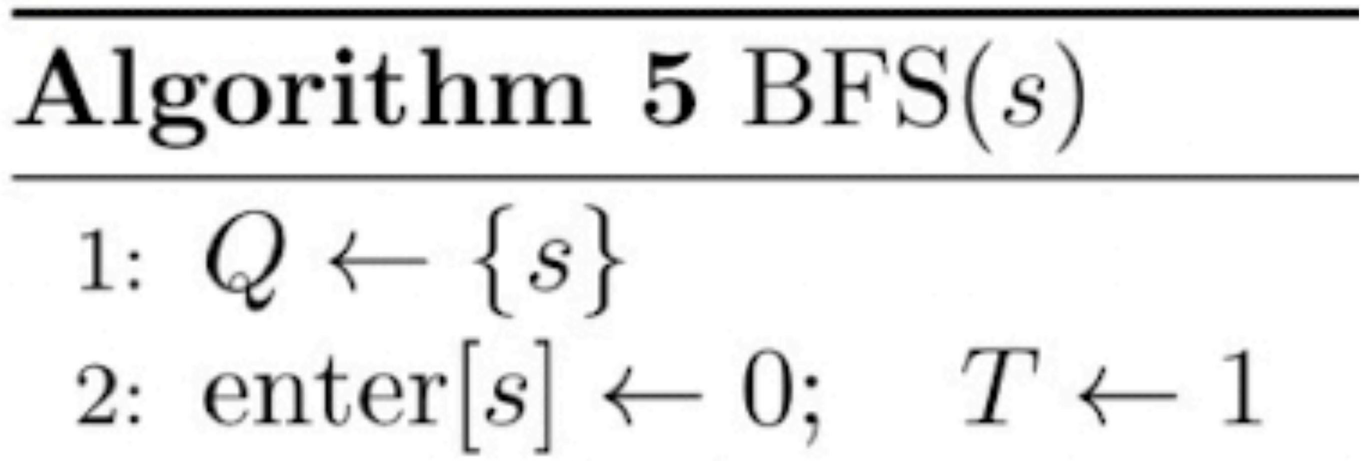
A	B	C	D	E	F	G	H
1	5	7			9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	

Graph Searchs

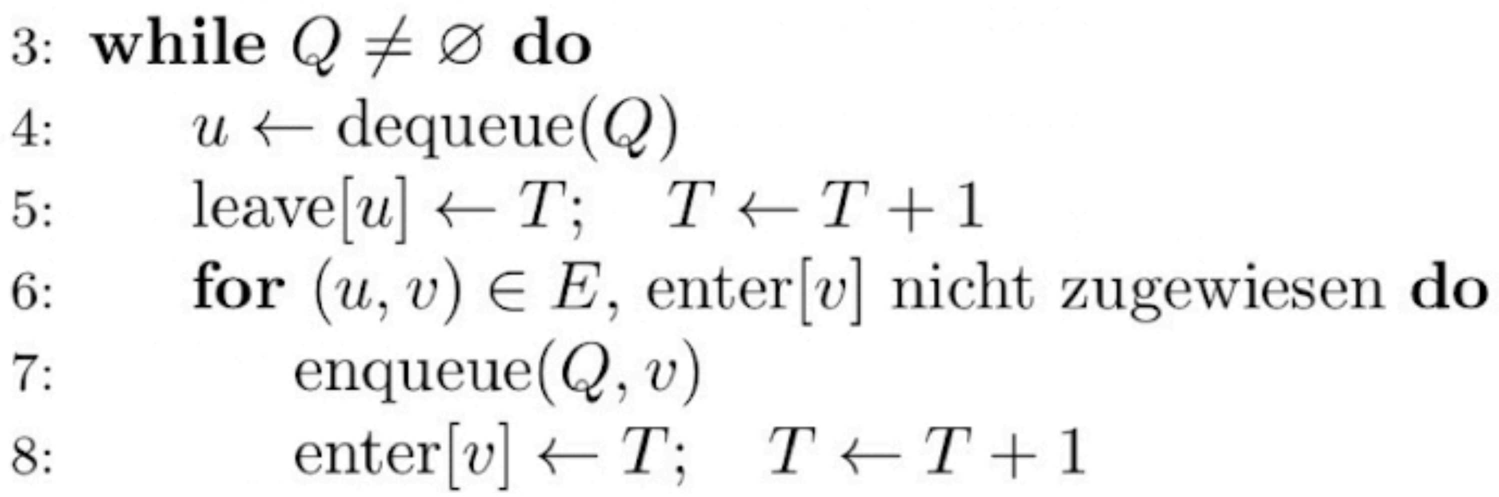
BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

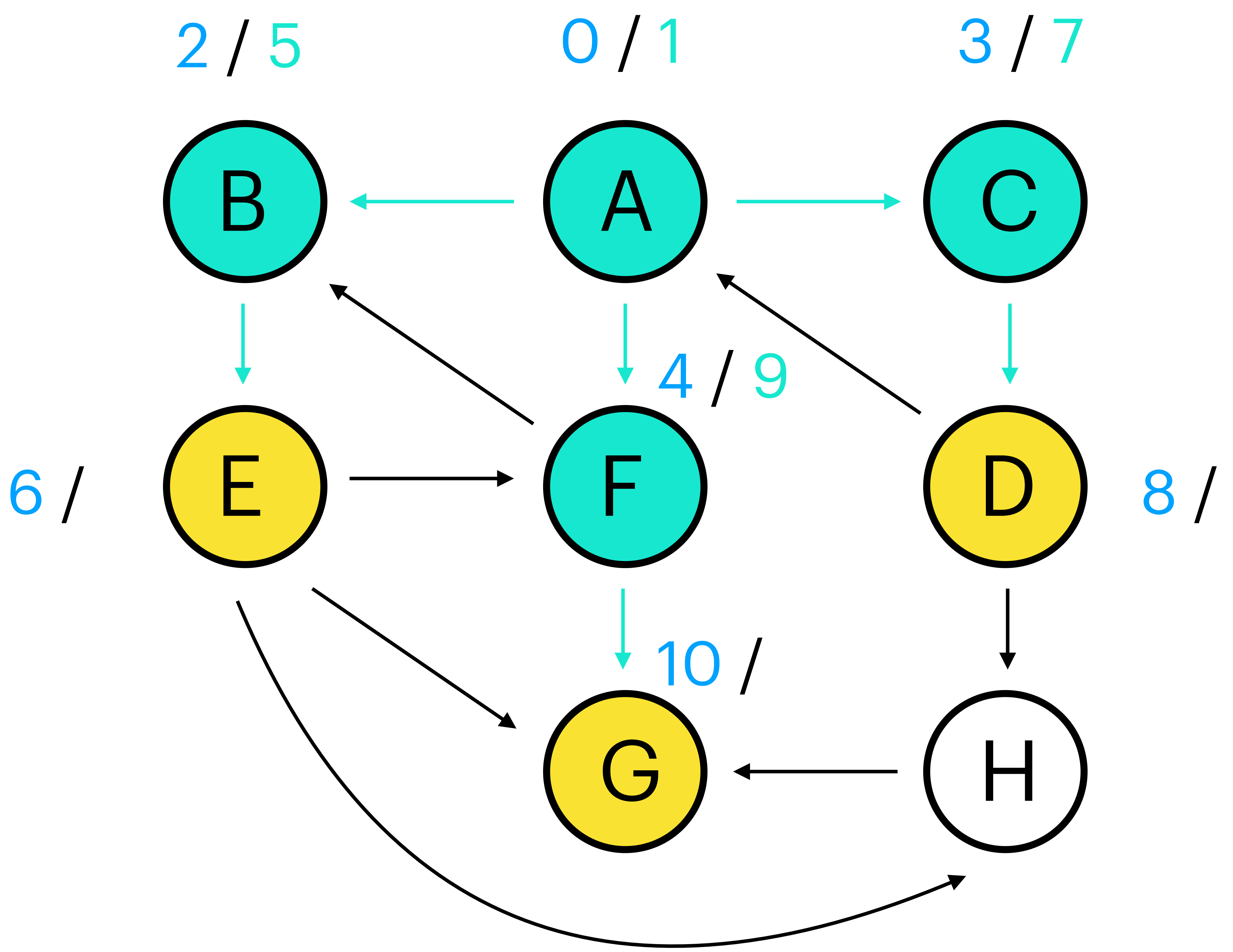
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : E - D - G



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	

leave[] :

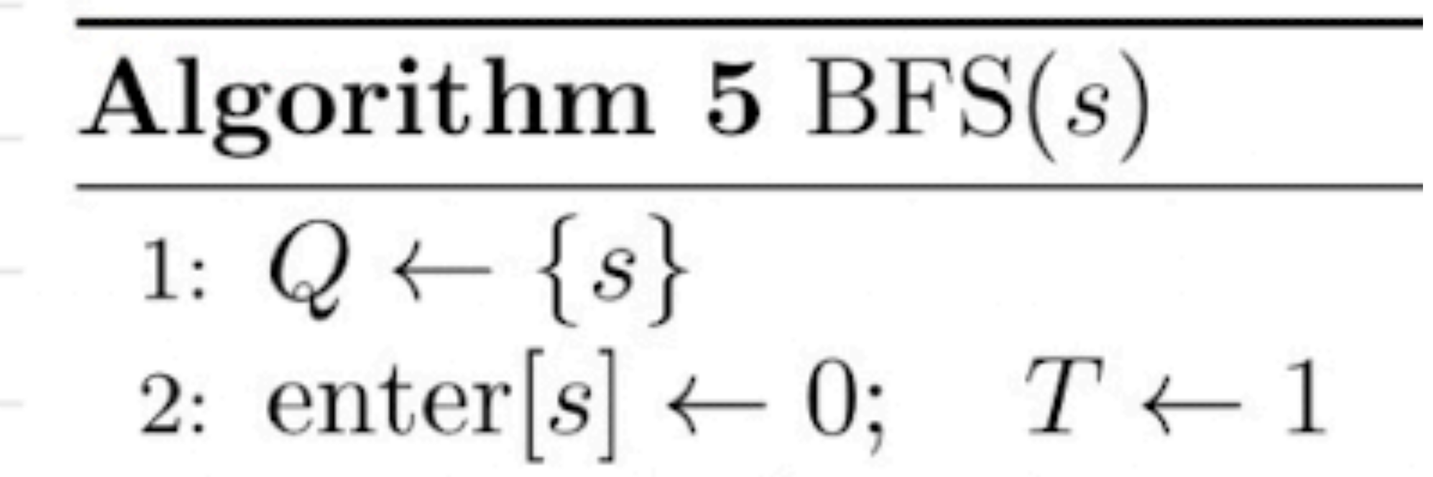
A	B	C	D	E	F	G	H
1	5	7			9		

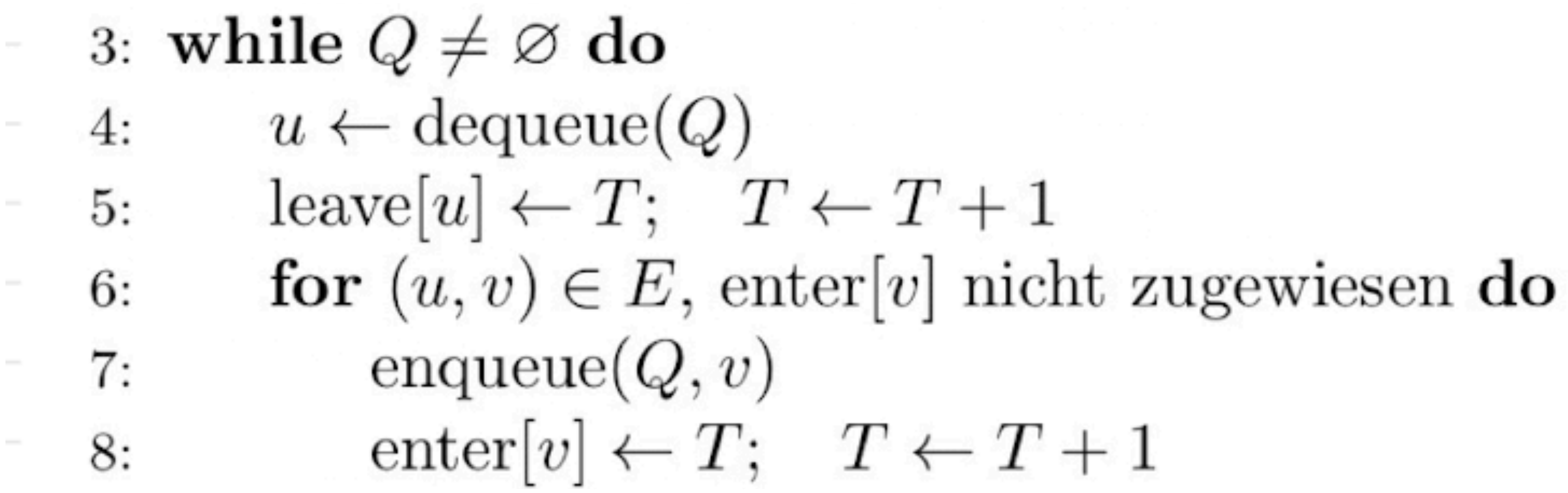
distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	

Graph Searchs

BFS - Example





```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : D - G

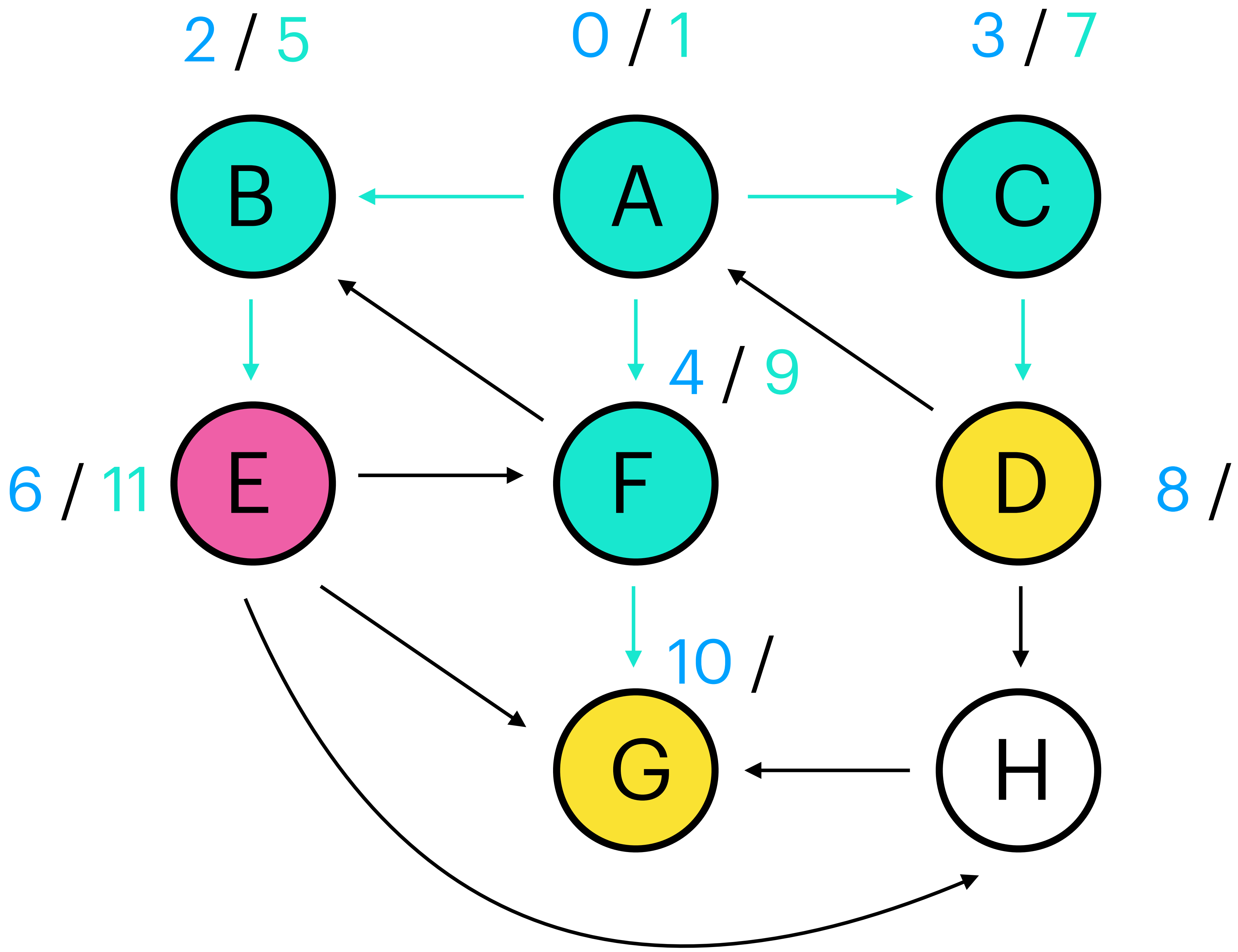
u = E

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	

leave[] :

A	B	C	D	E	F	G	H
1	5	7		11	9		

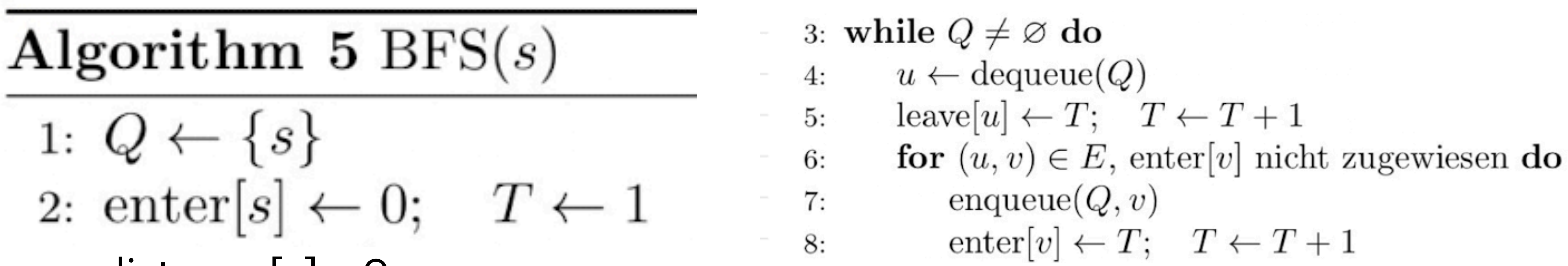


distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	

Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : D - G

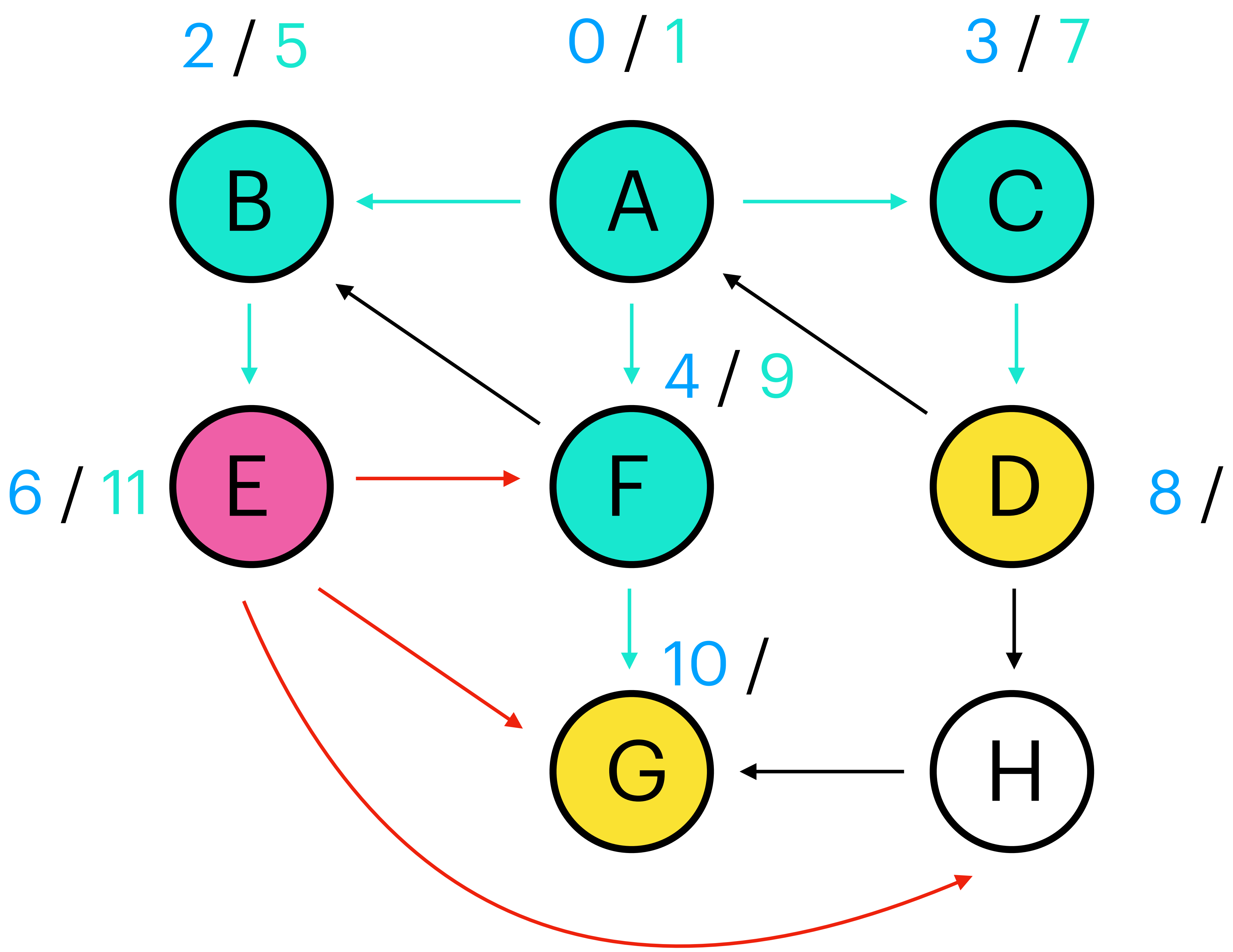
u = E

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	

leave[] :

A	B	C	D	E	F	G	H
1	5	7		11	9		

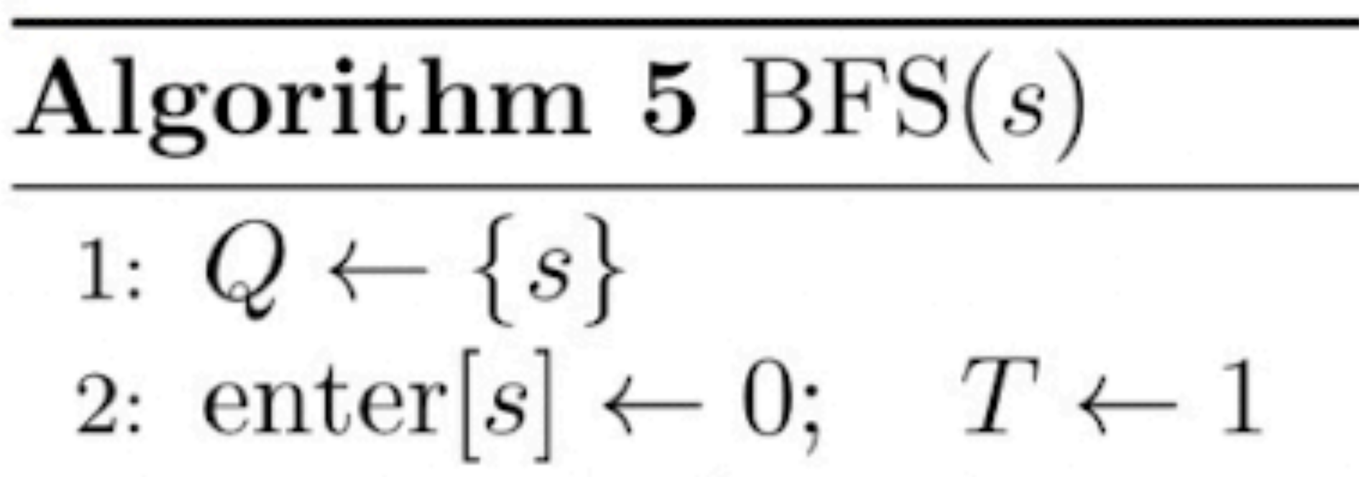


distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	

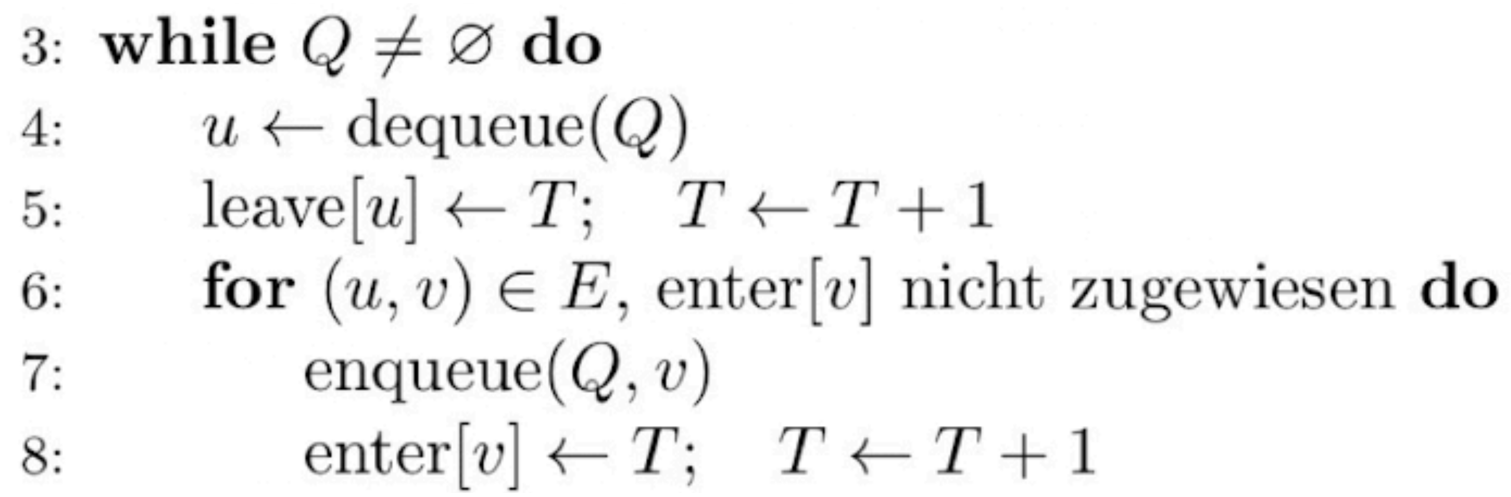
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : D - G - H

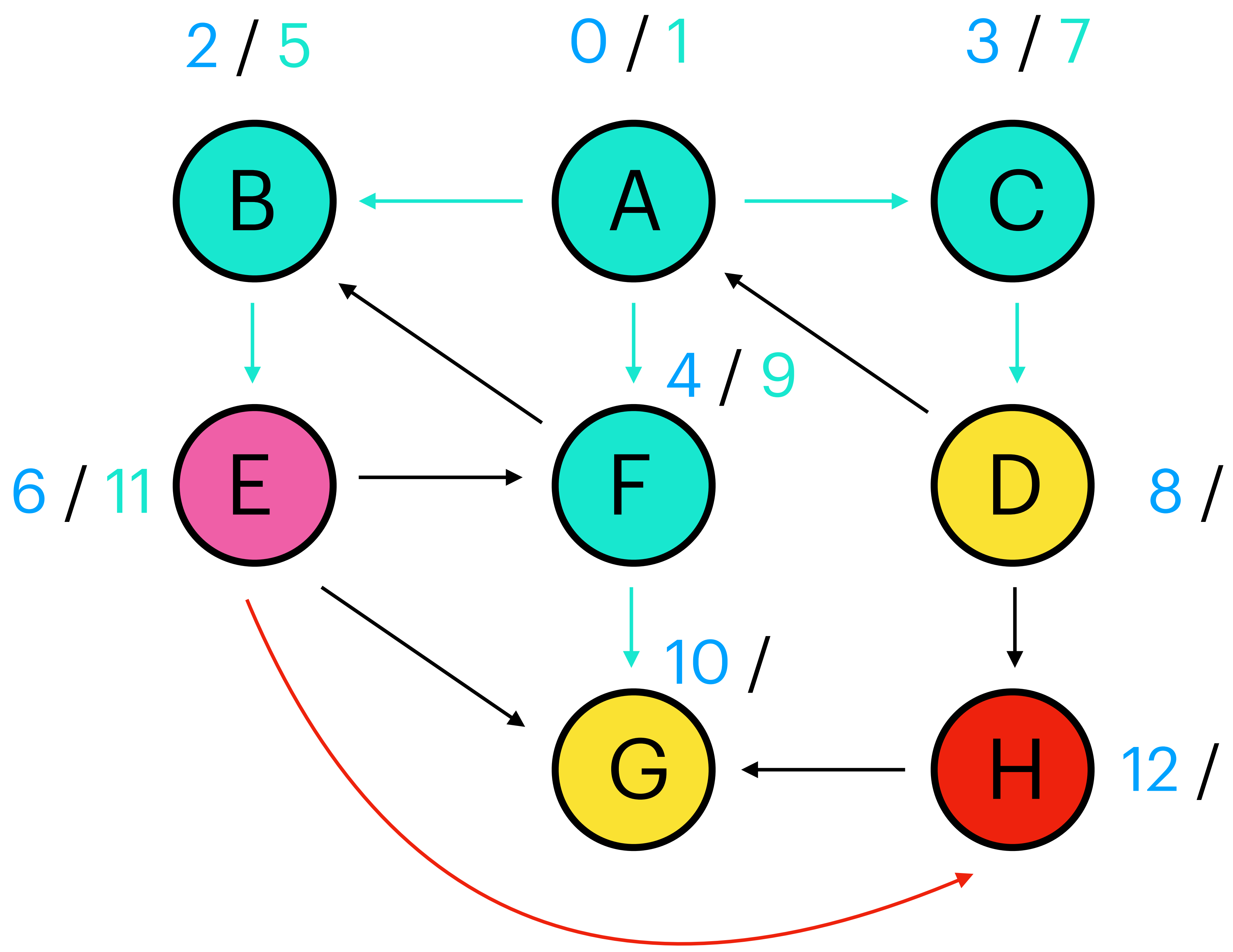
u = E

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

A	B	C	D	E	F	G	H
1	5	7		11	9		

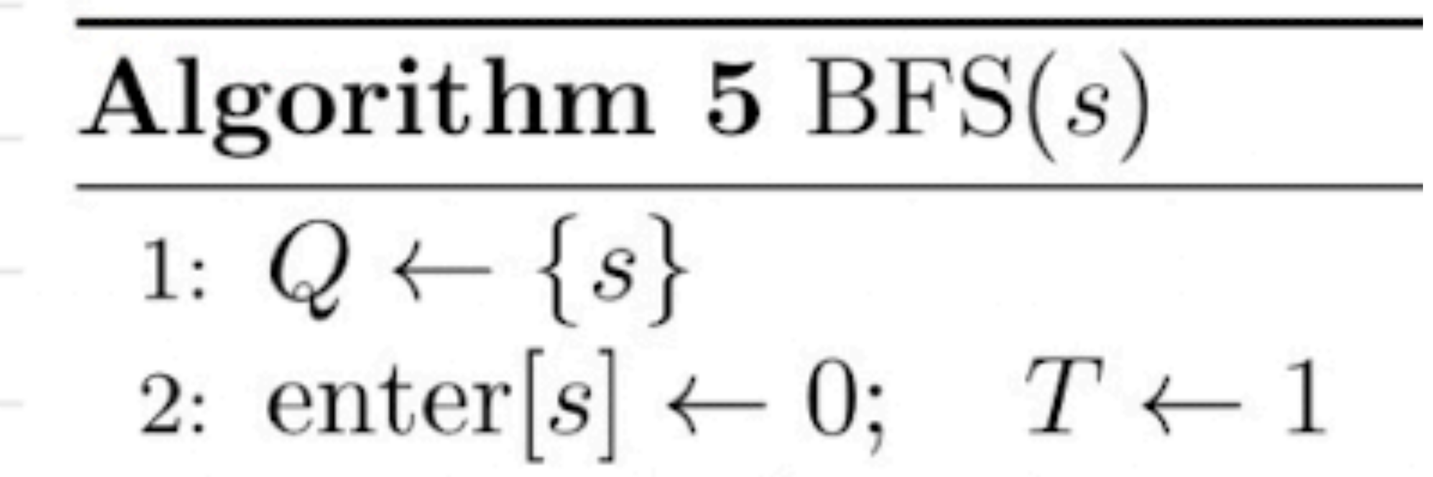


distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

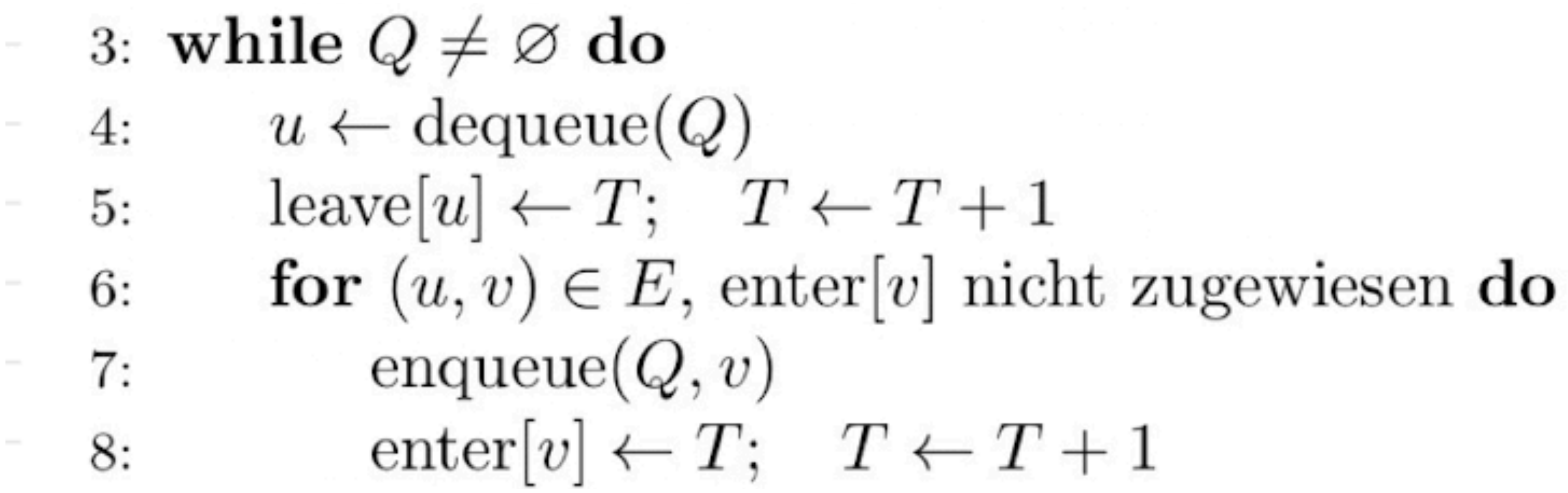
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
```

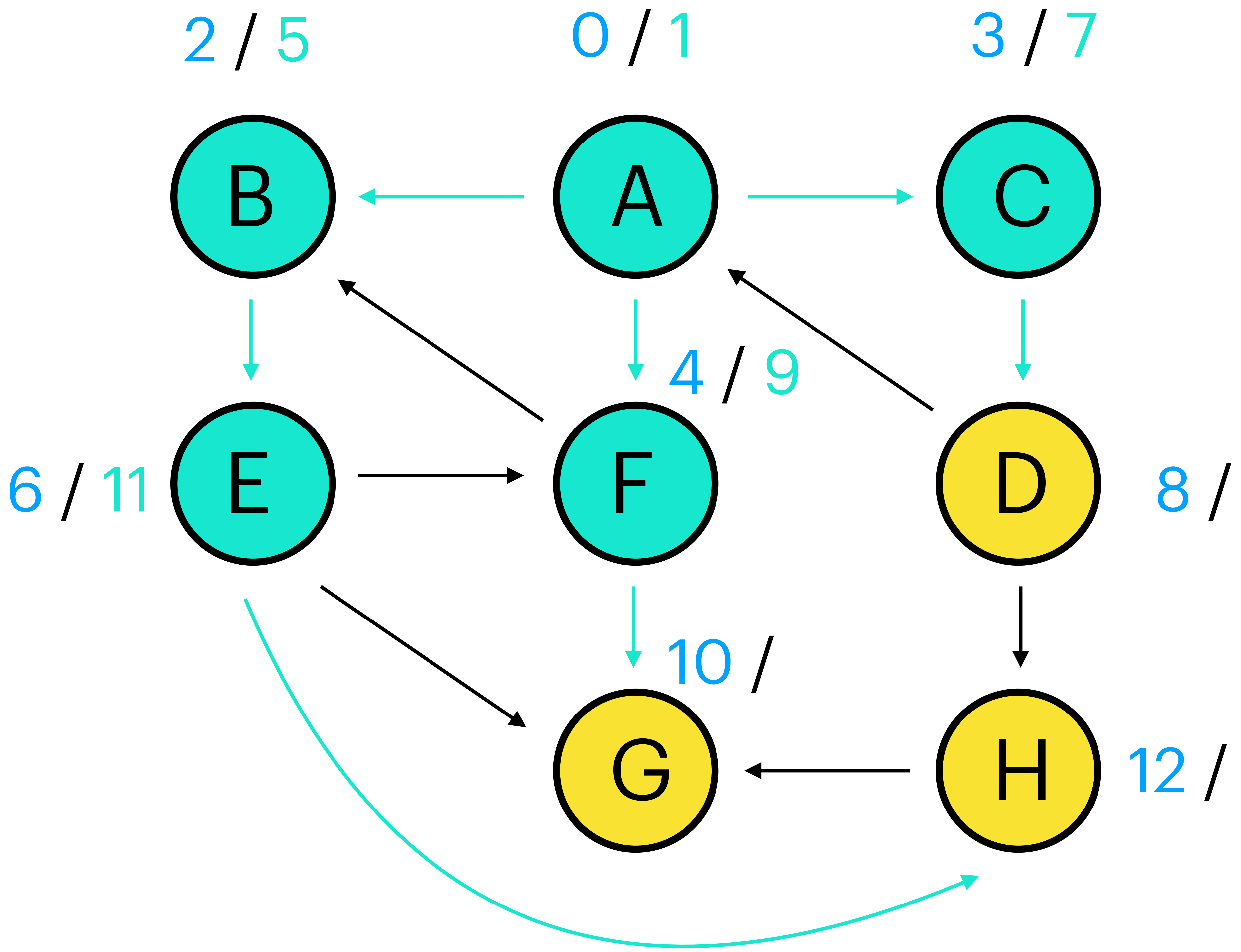
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : D - G - H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

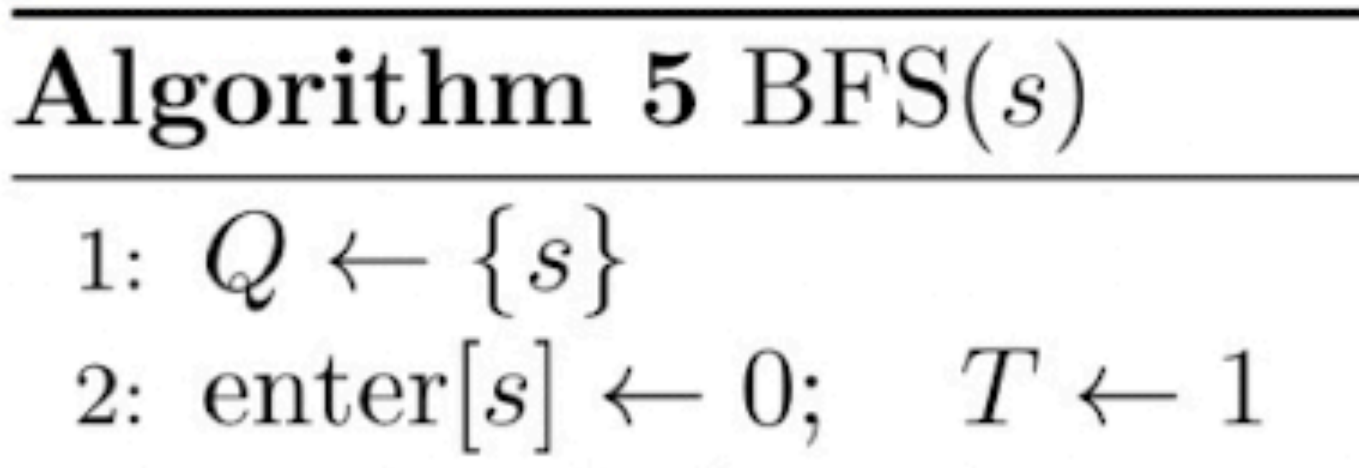
A	B	C	D	E	F	G	H
1	5	7		11	9		

distance[] :

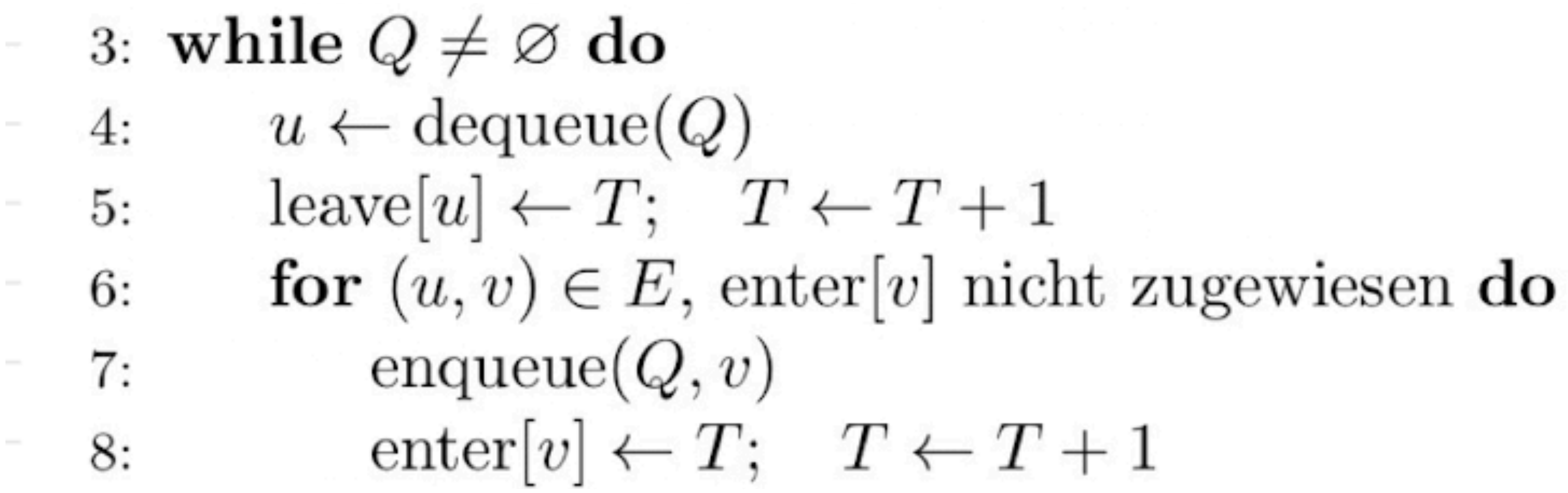
A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

Graph Searchs

BFS - Example

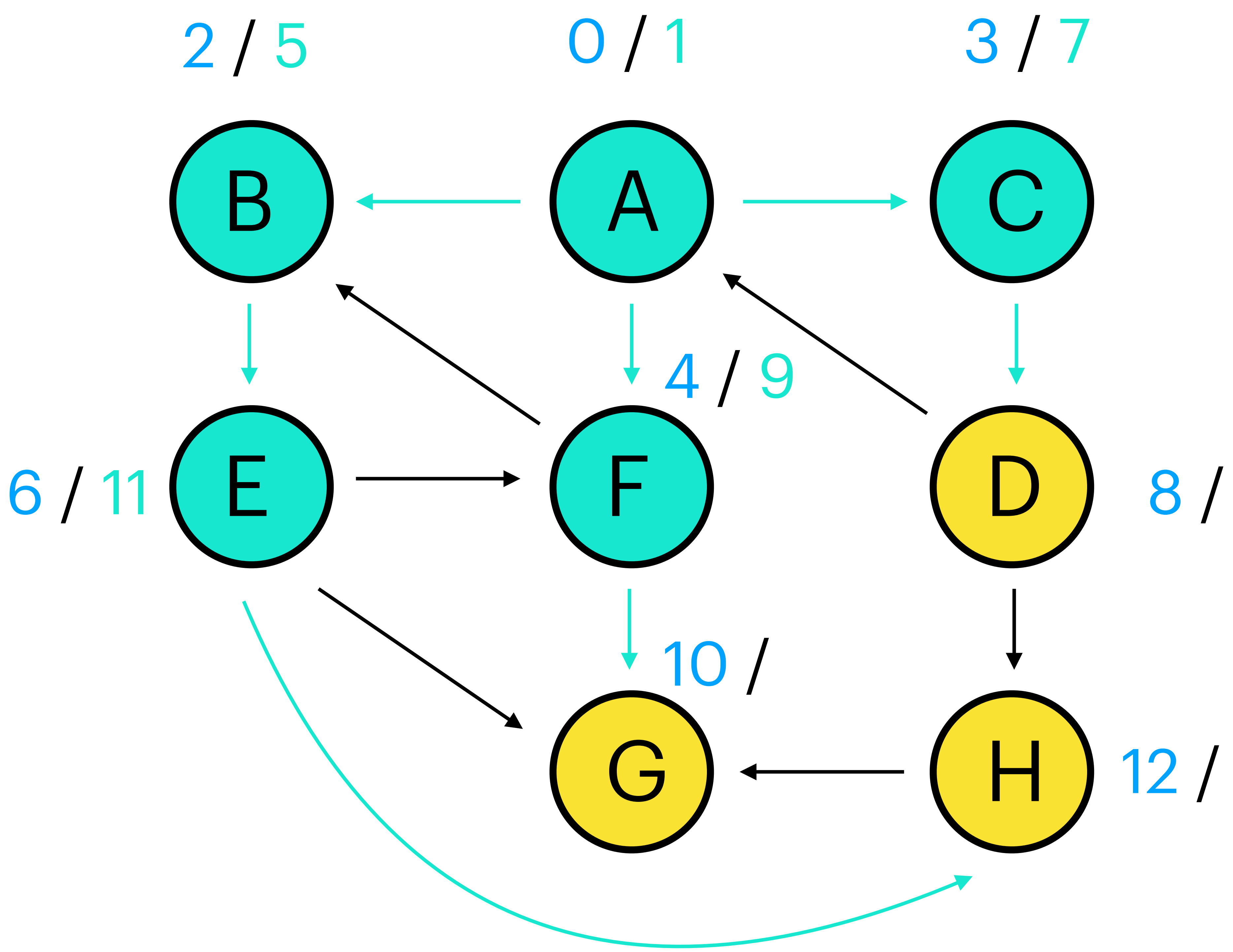


```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T+1$
    for $(u,v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T+1$
        distance[v] <- distance[u] + 1 ;
```

Q : D - G - H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

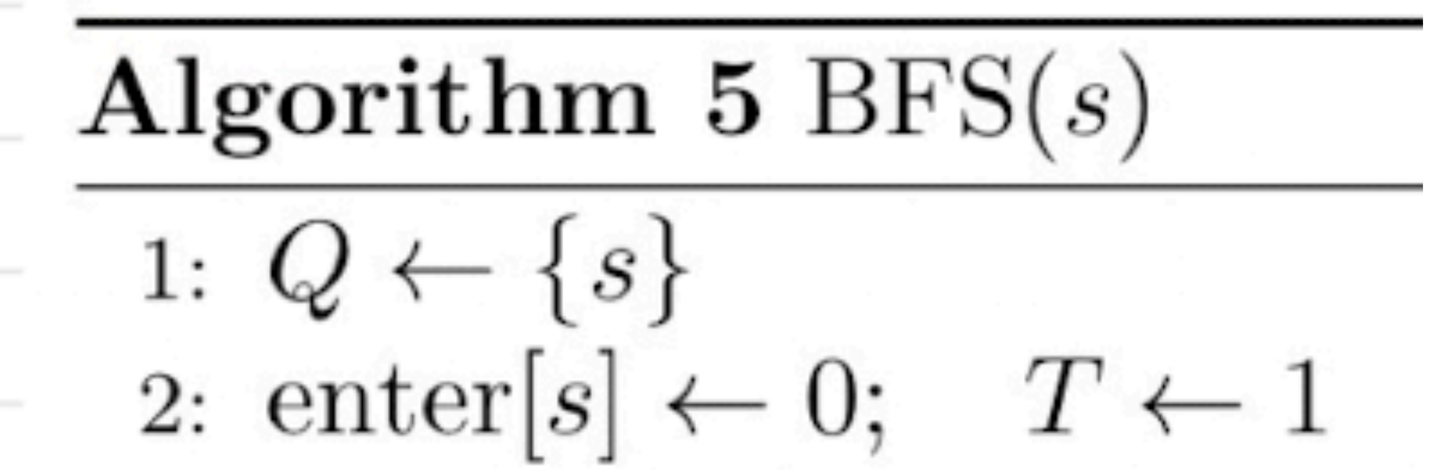
A	B	C	D	E	F	G	H
1	5	7		11	9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

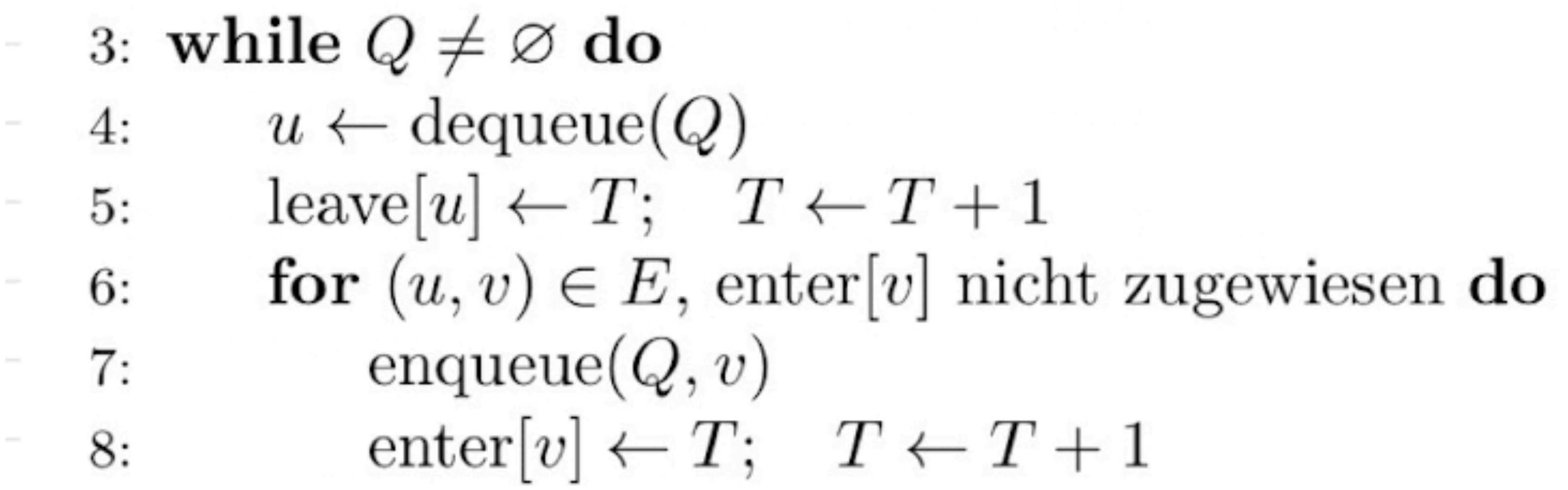
Graph Searchs

BFS - Example



Algorithm 5 BFS(s)

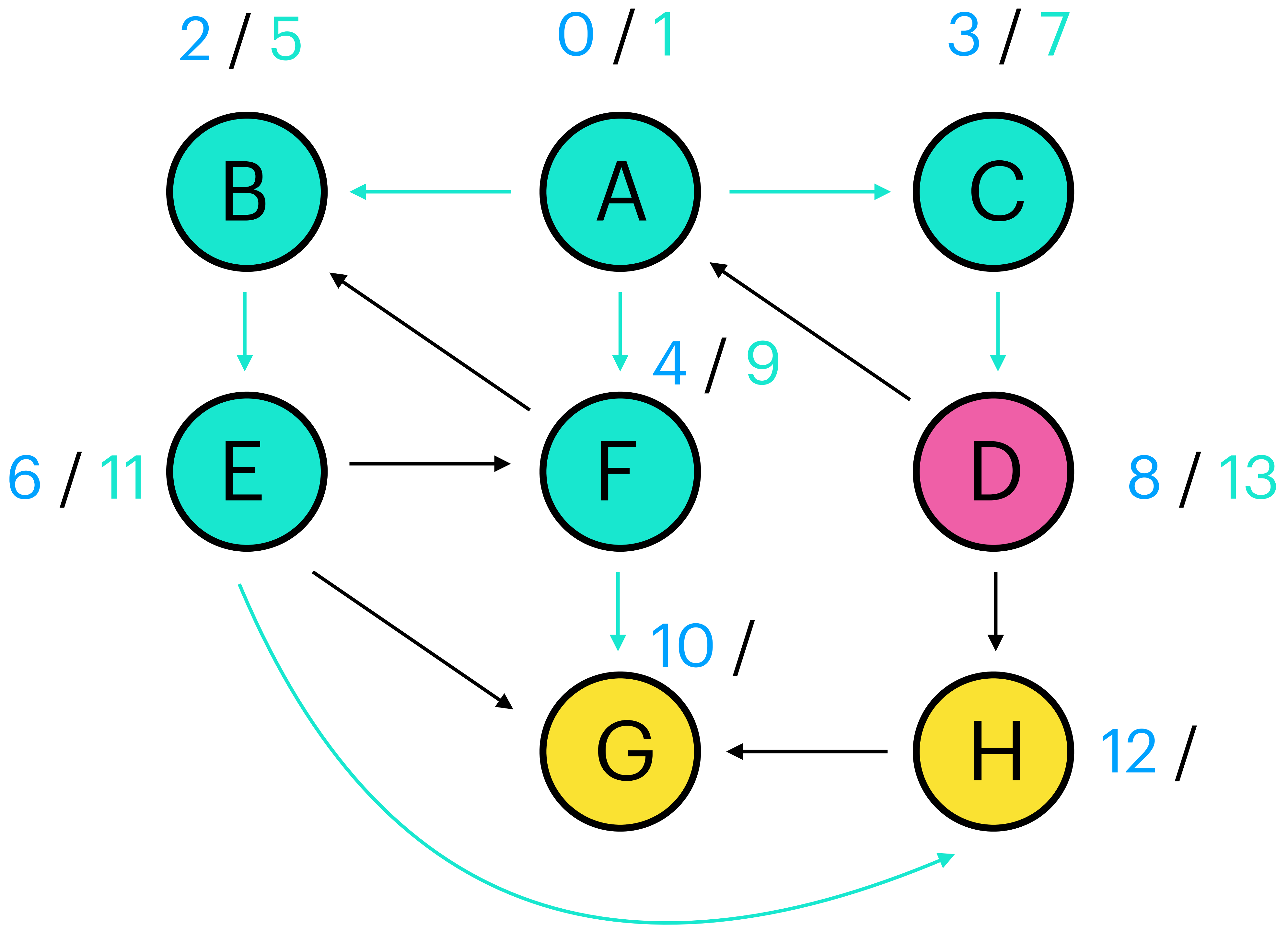
```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : G - H

u = D



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

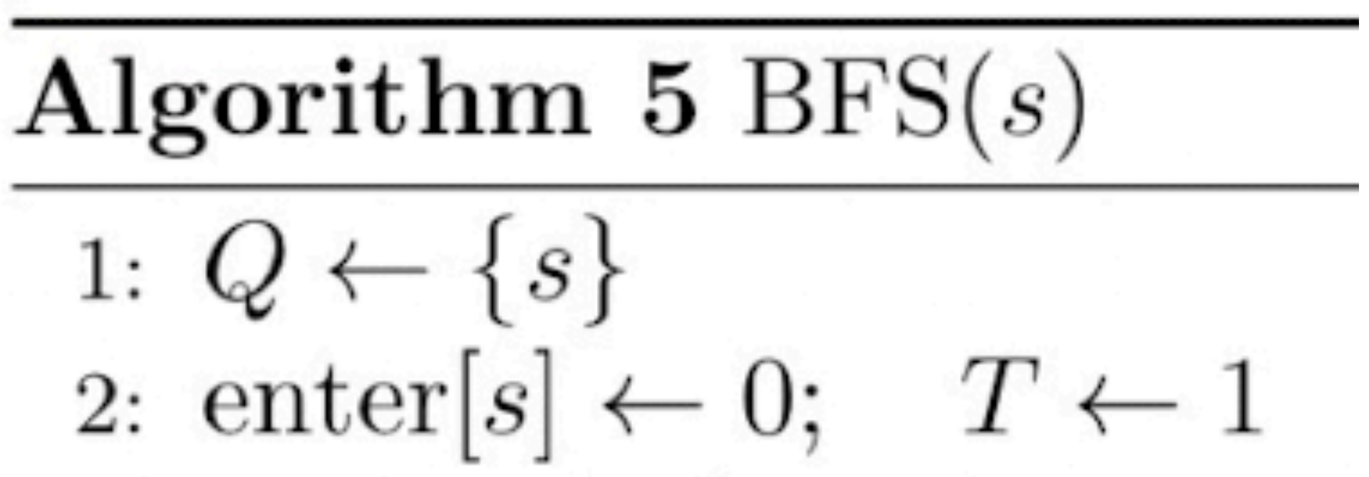
A	B	C	D	E	F	G	H
1	5	7	13	11	9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

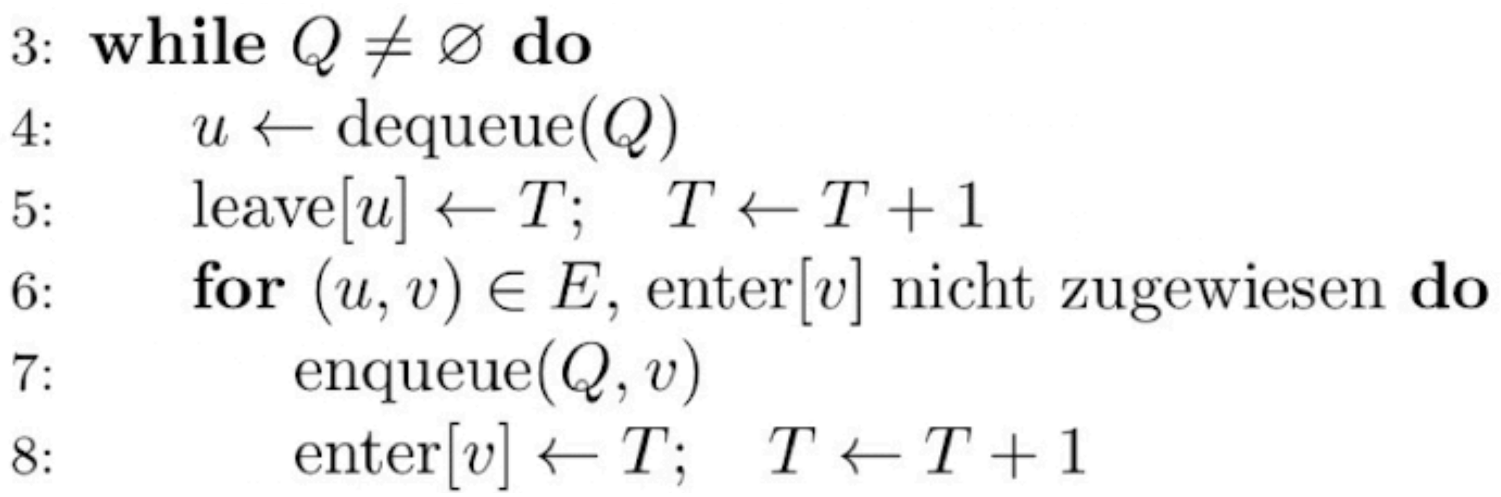
Graph Searchs

BFS - Example



```
Algorithm 5 BFS(s)
Q ← {s}
enter[s] ← 0;   T ← 1
```

distance[s] = 0 ;

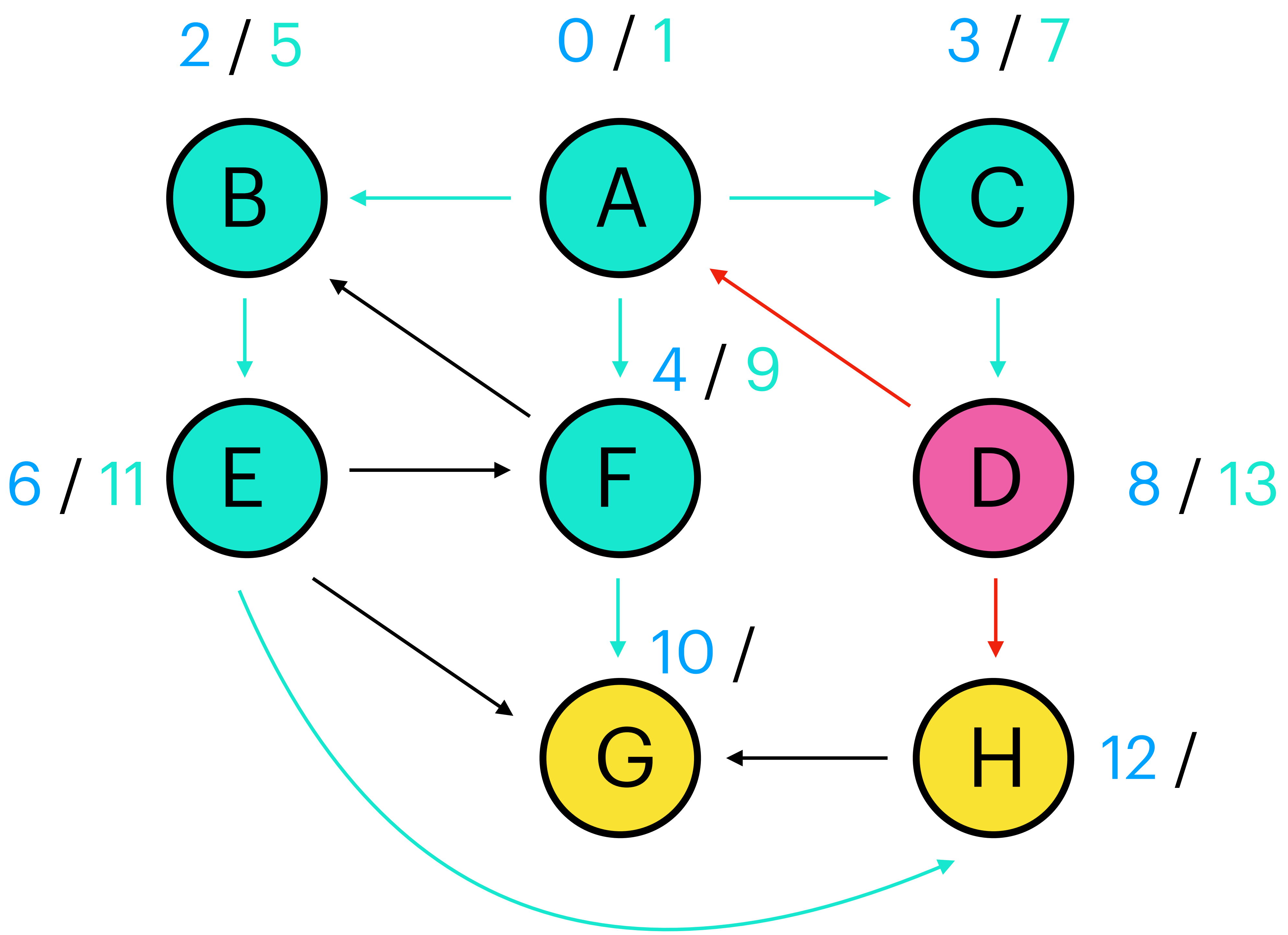


```
while Q ≠ ∅ do
    u ← dequeue(Q)
    leave[u] ← T;   T ← T + 1
    for (u, v) ∈ E, enter[v] nicht zugewiesen do
        enqueue(Q, v)
        enter[v] ← T;   T ← T + 1
```

distance[v] <- distance[u] + 1 ;

Q : G - H

u = D



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

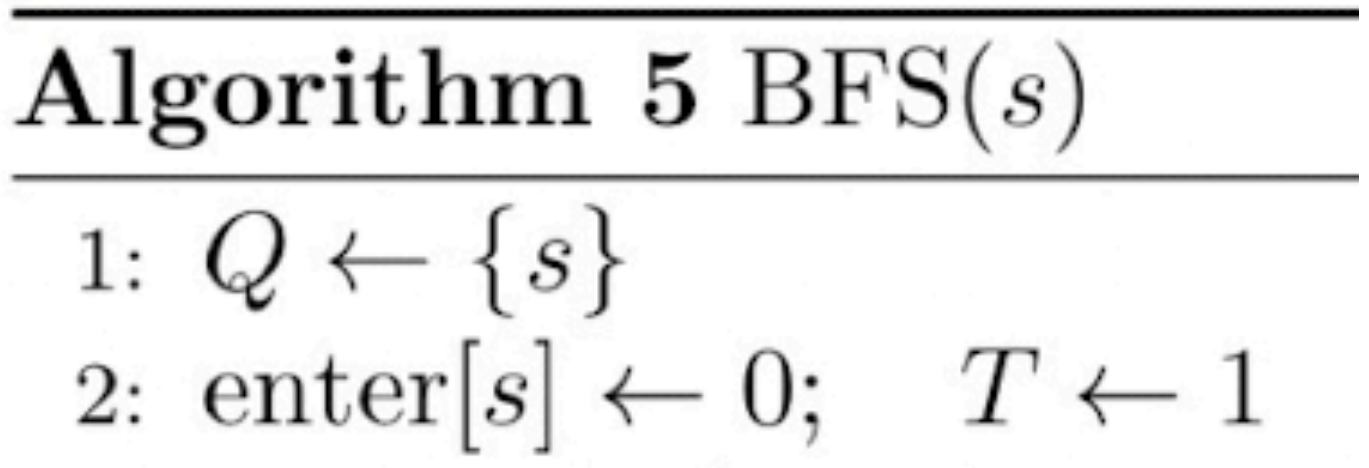
A	B	C	D	E	F	G	H
1	5	7	13	11	9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

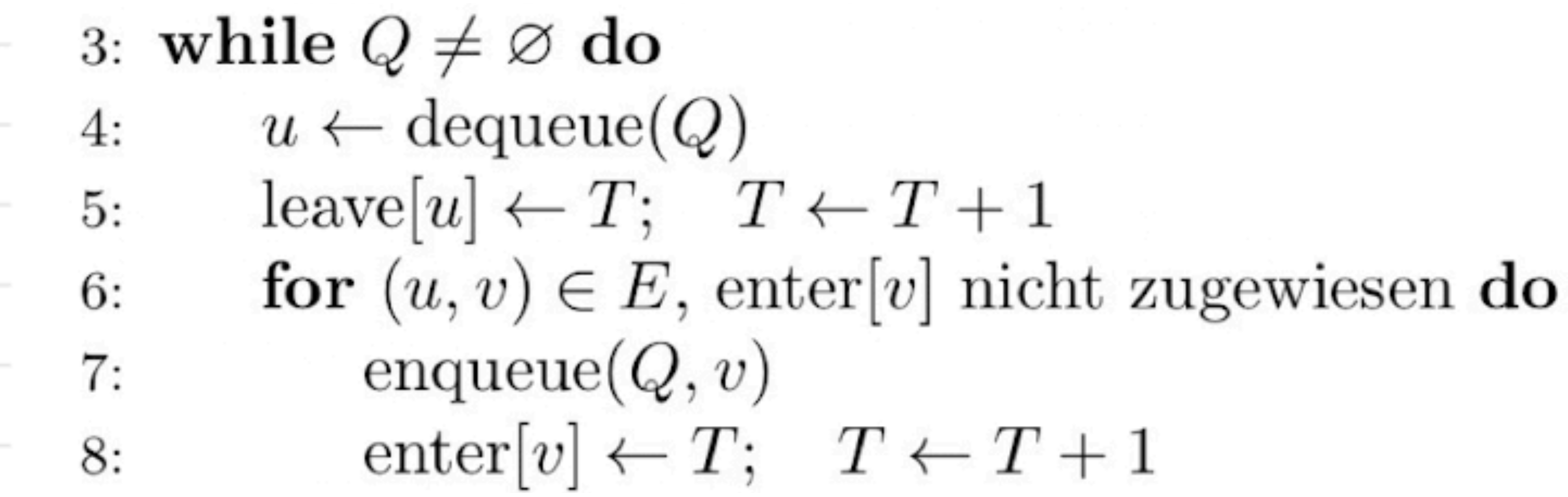
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

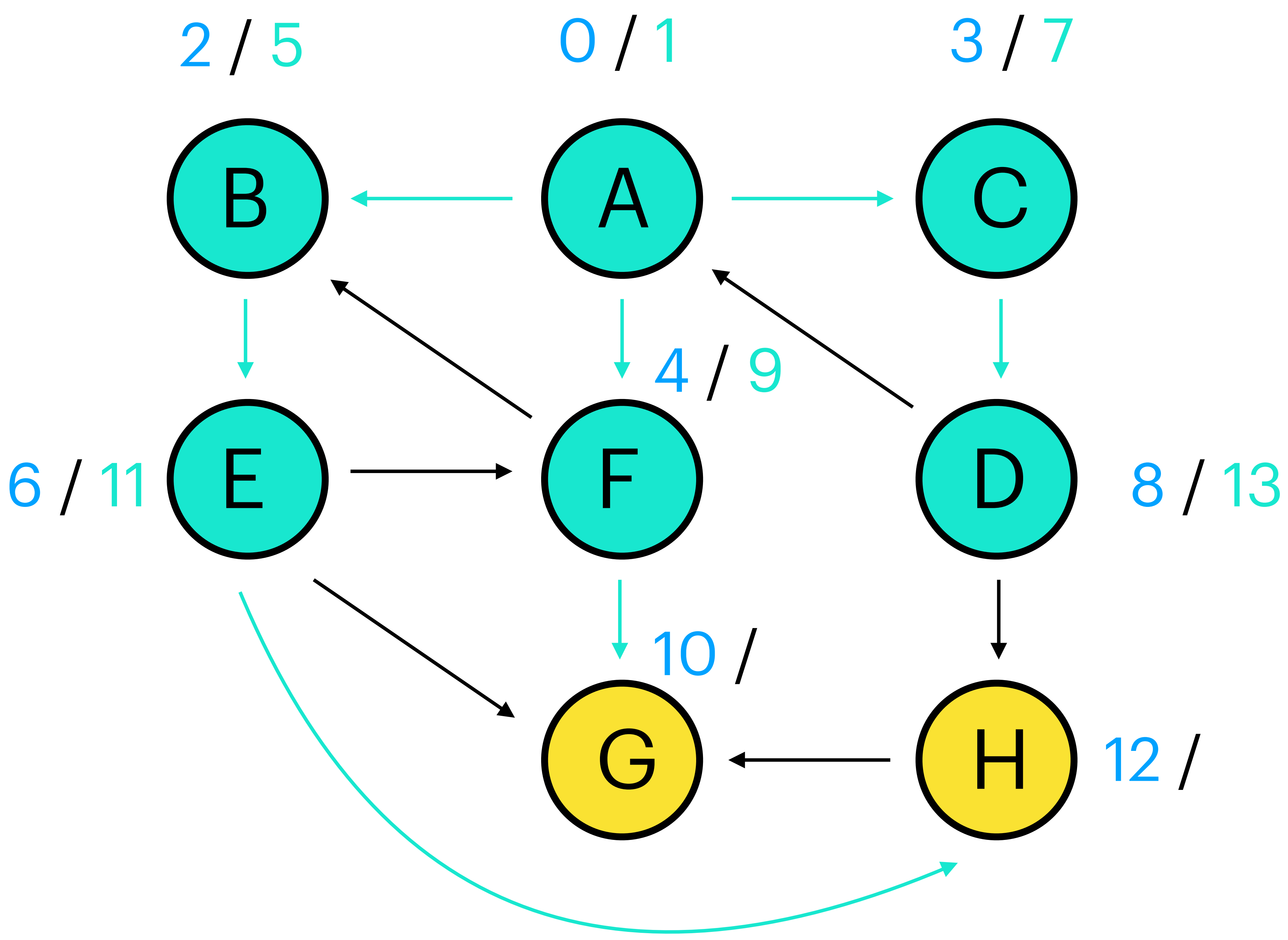
distance[s] = 0 ;



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q : G - H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

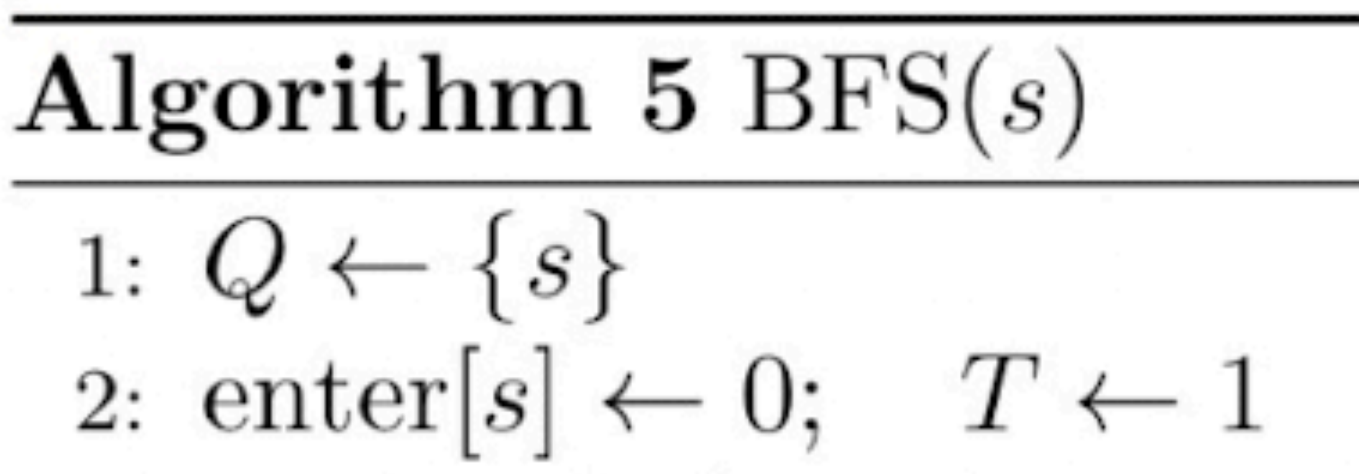
A	B	C	D	E	F	G	H
1	5	7	13	11	9		

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

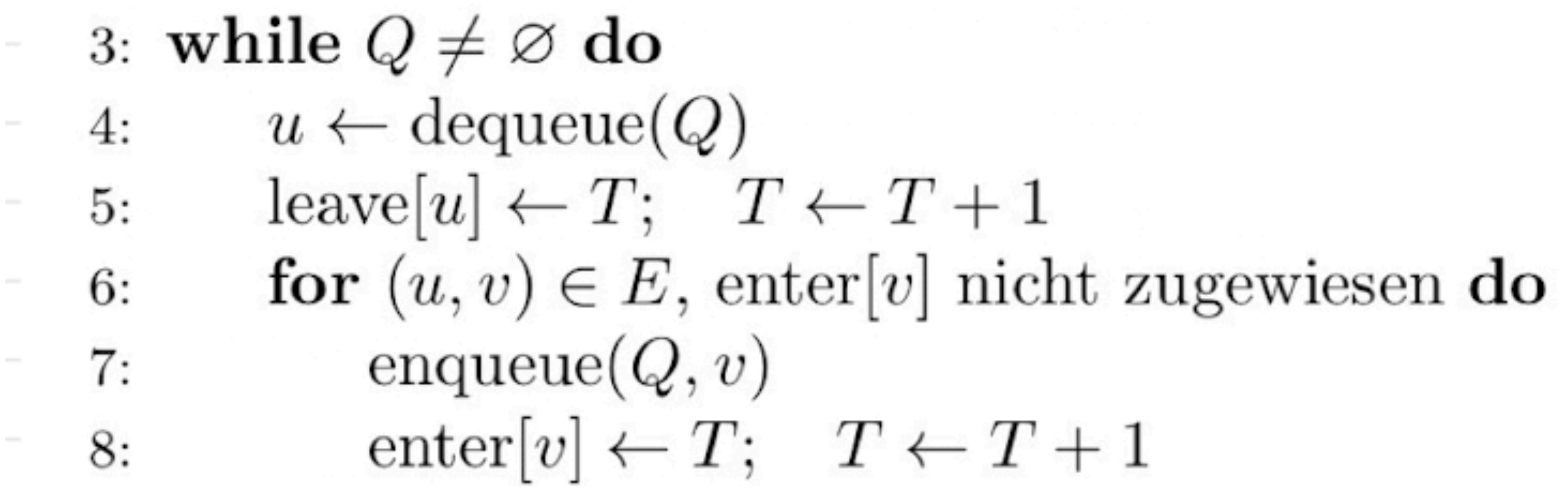
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

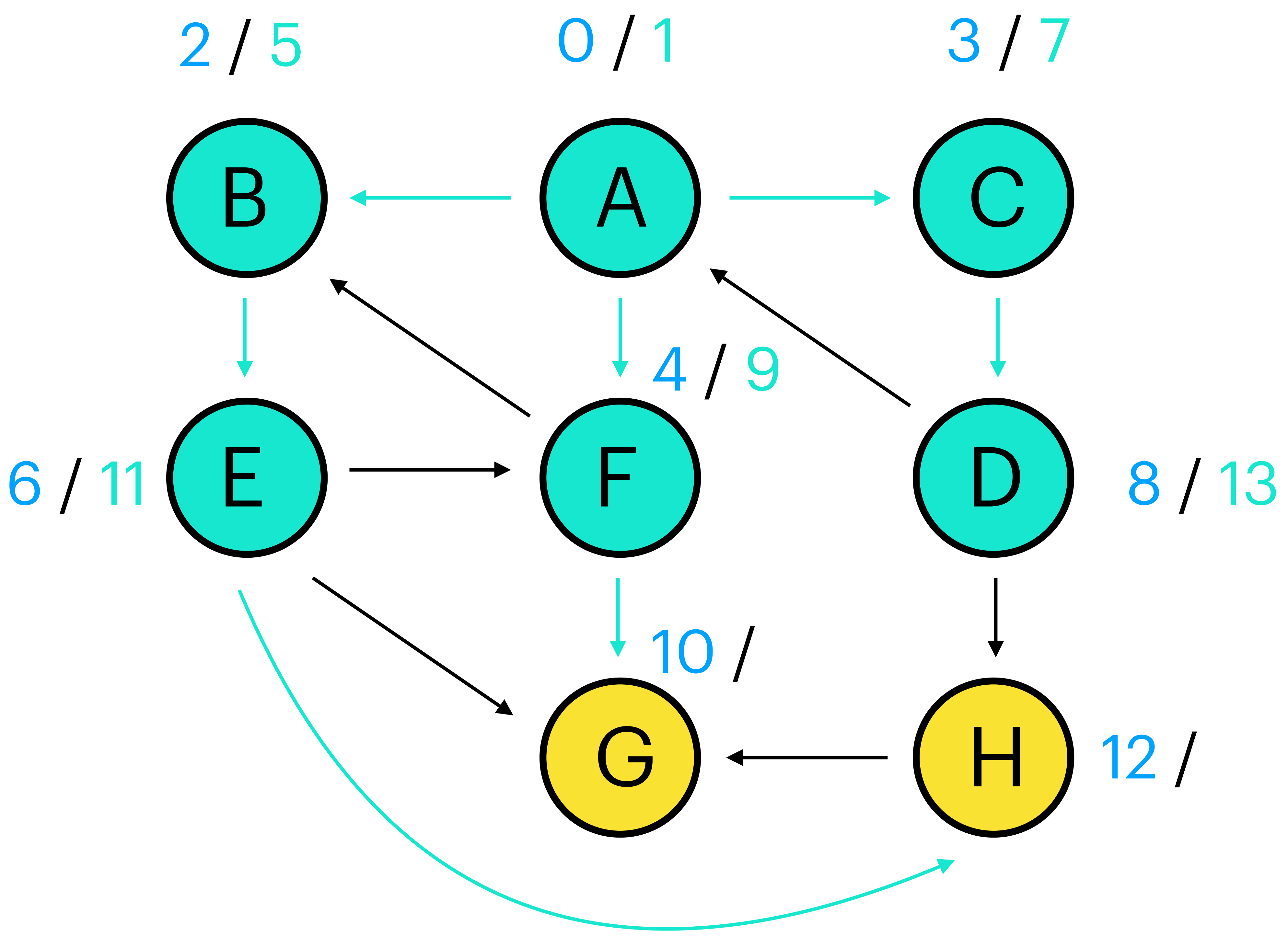
```
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

```
        distance[v] <- distance[u] + 1 ;
```

Q : G - H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

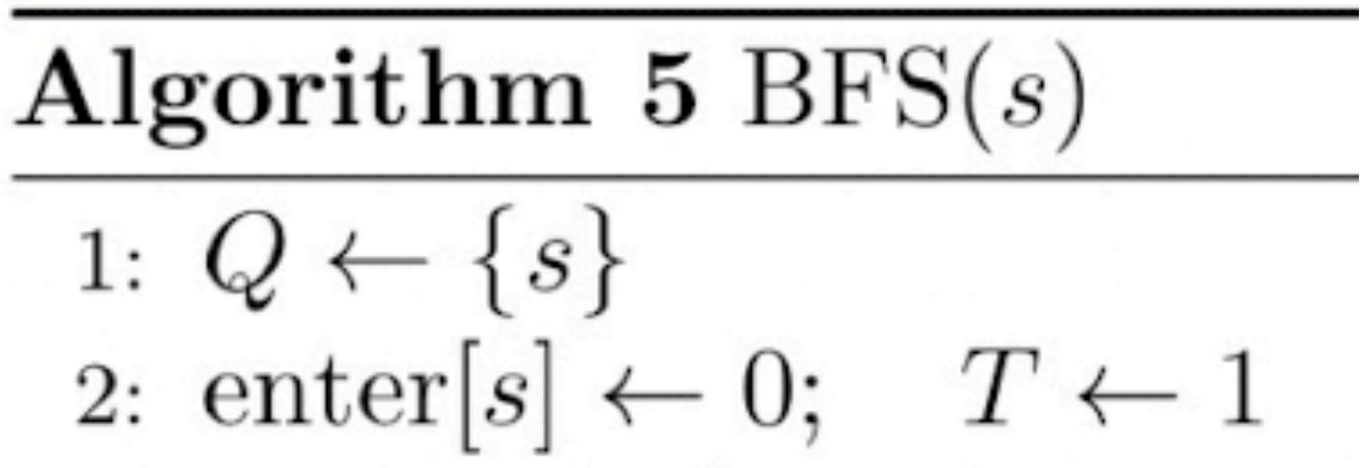
A	B	C	D	E	F	G	H
1	5	7	13	11	9		

distance[] :

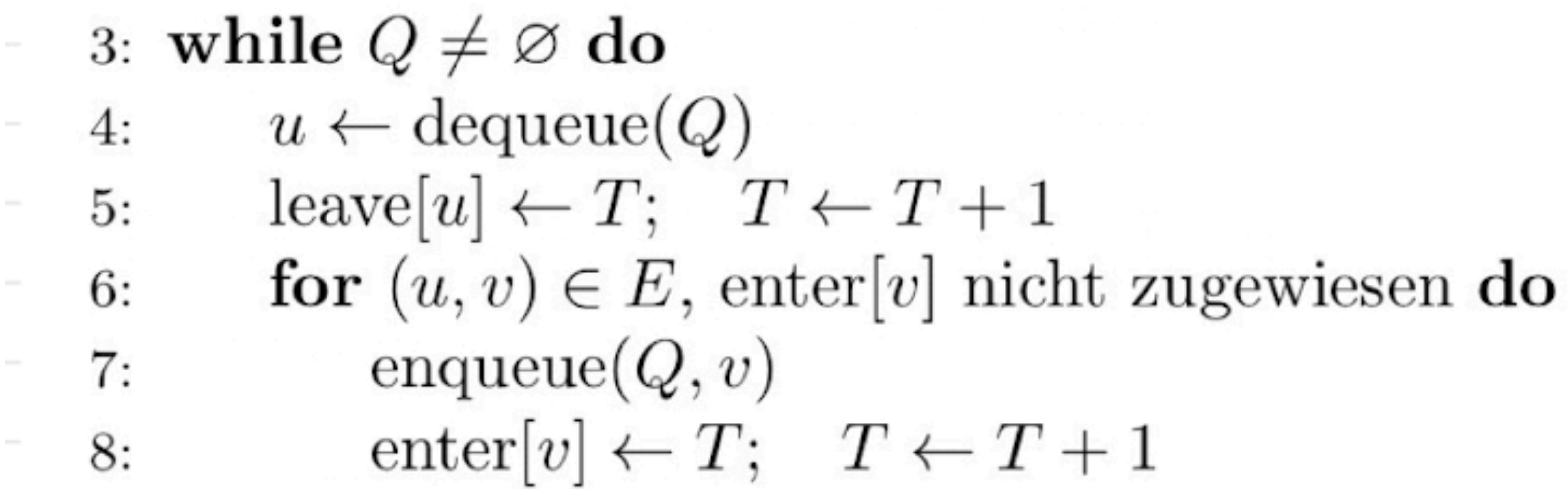
A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : H

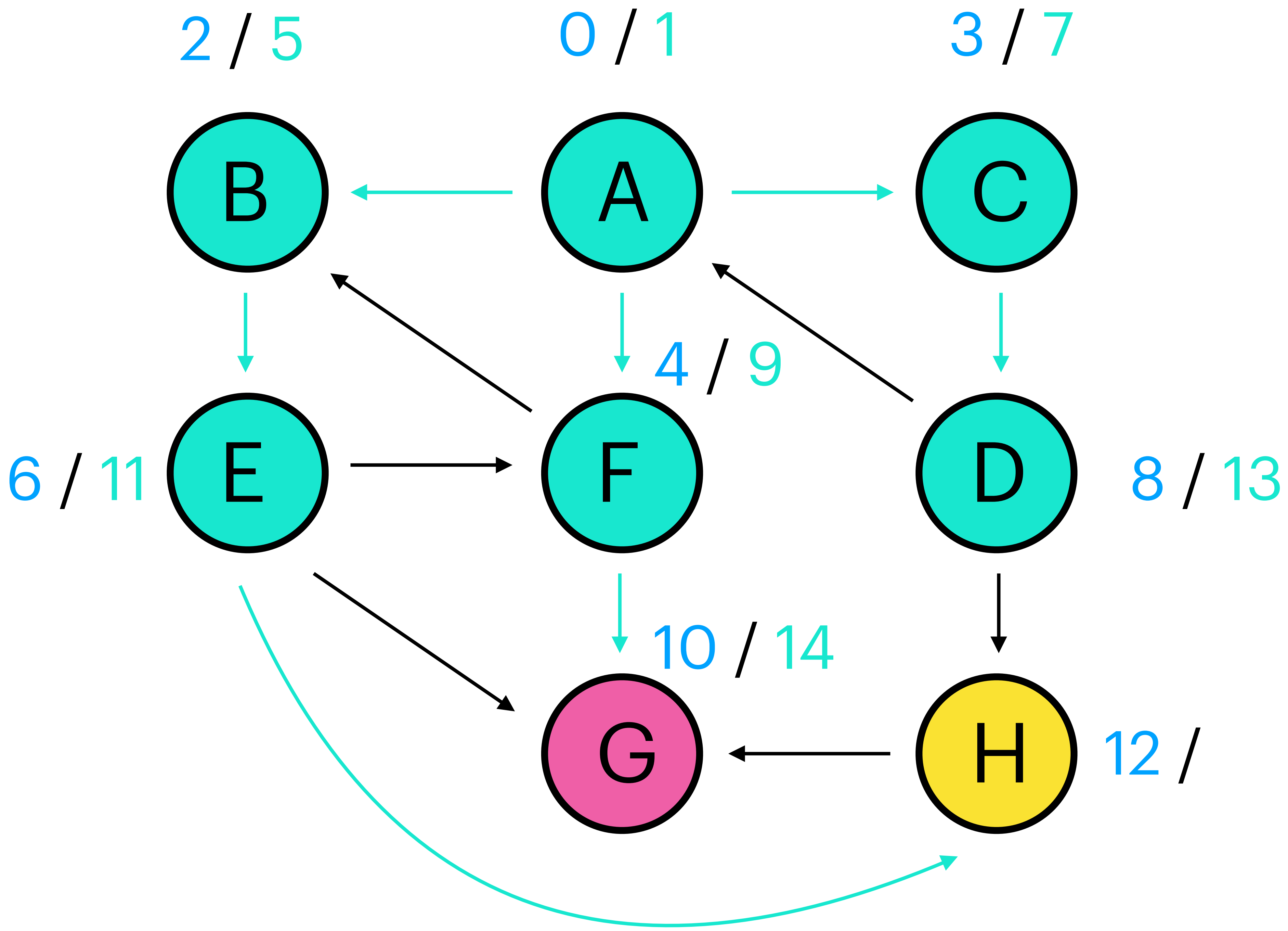
u = G

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	

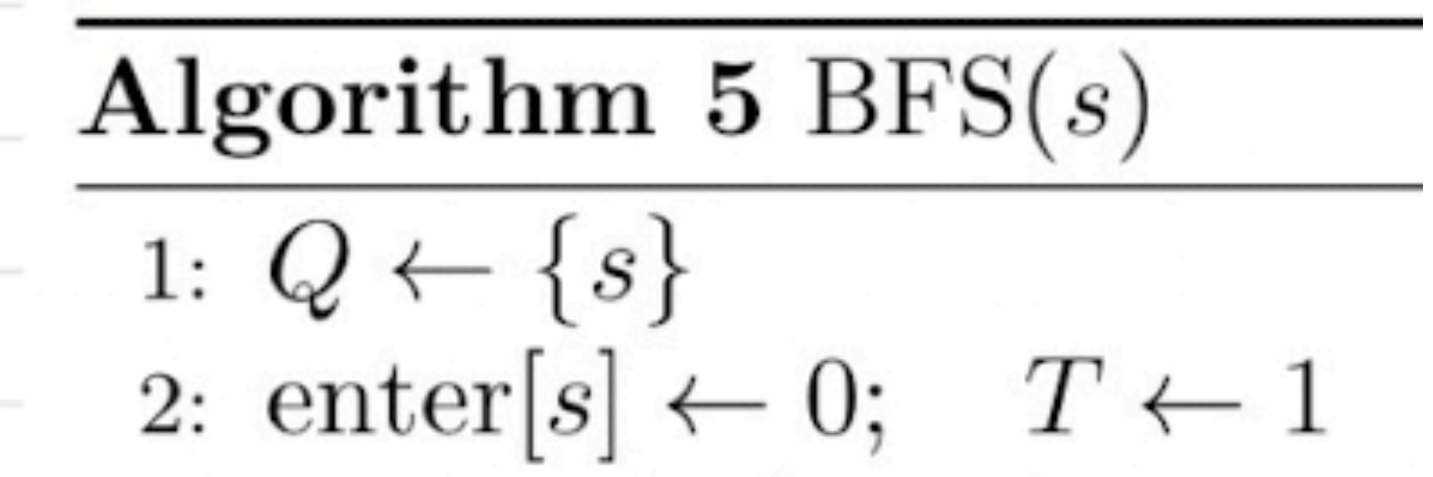


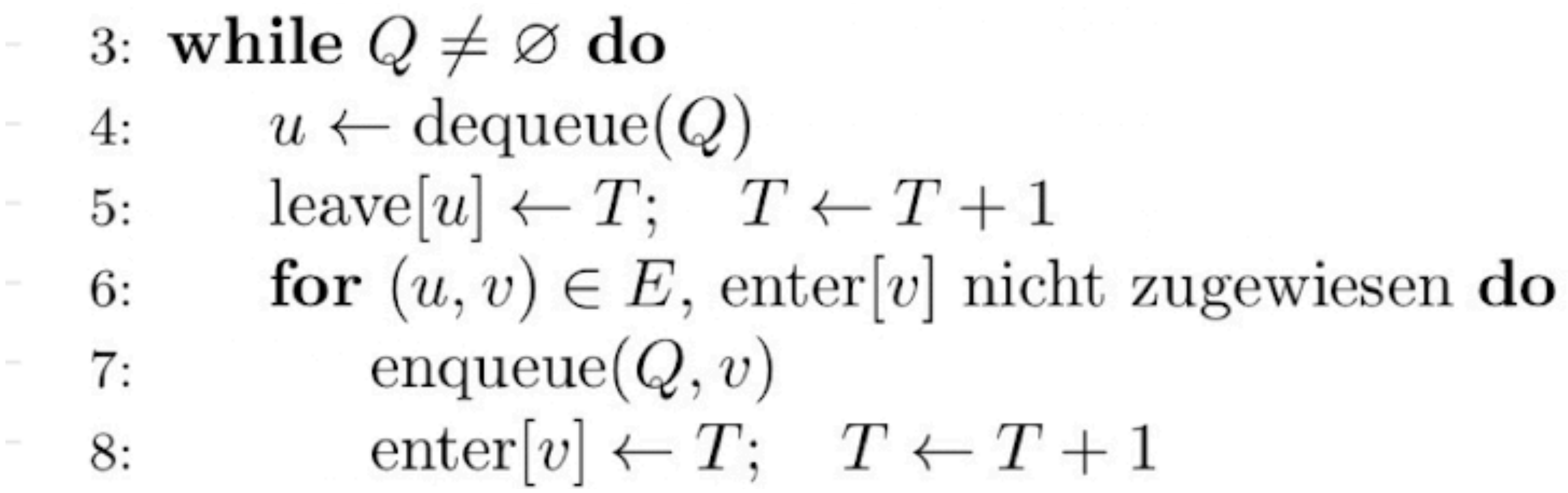
distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

Graph Searchs

BFS - Example

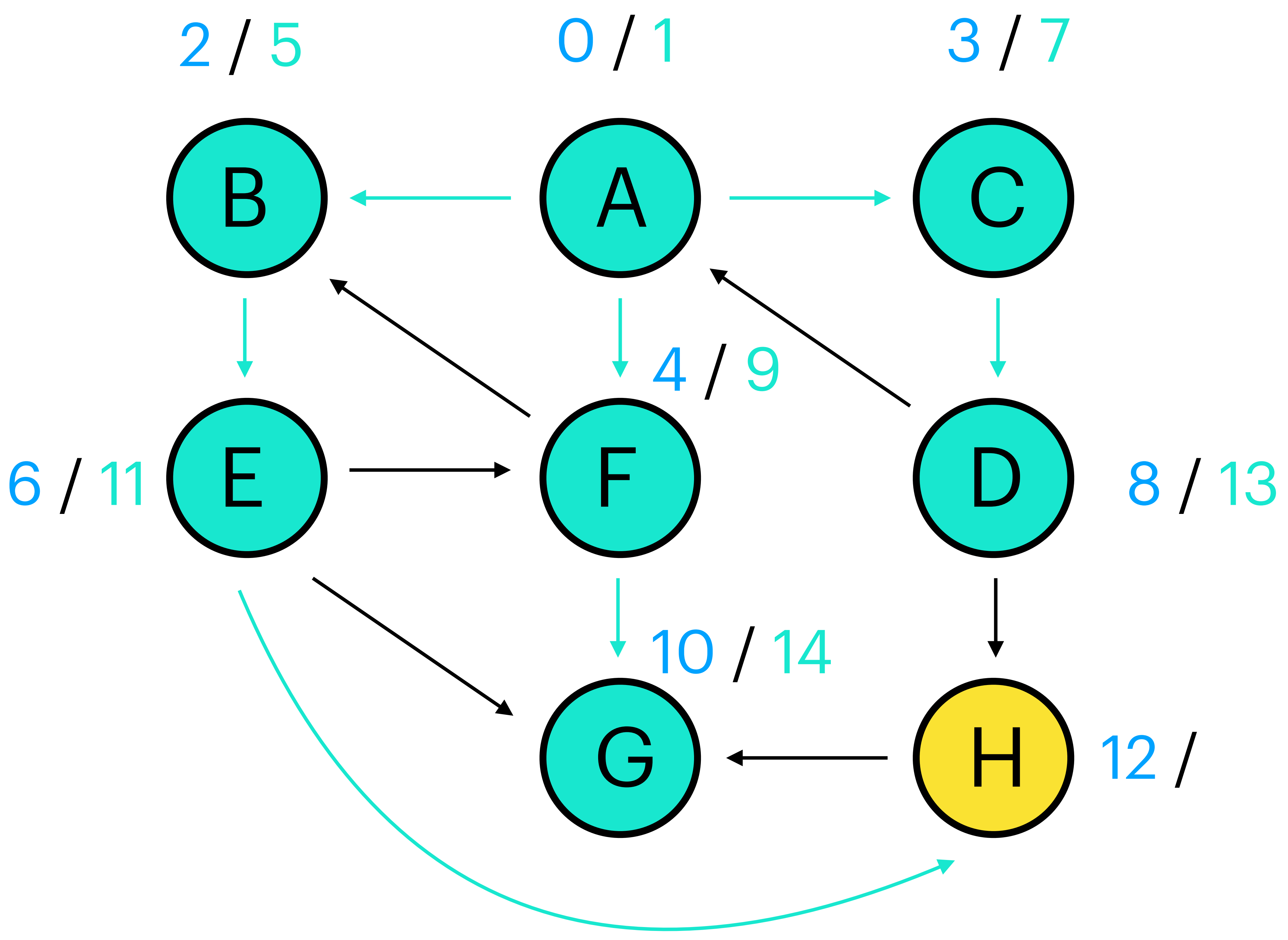




Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
$\text{enter}[s] \leftarrow 0; \quad T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow \text{dequeue}(Q)$
    $\text{leave}[u] \leftarrow T; \quad T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        $\text{enqueue}(Q, v)$
        $\text{enter}[v] \leftarrow T; \quad T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q : H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

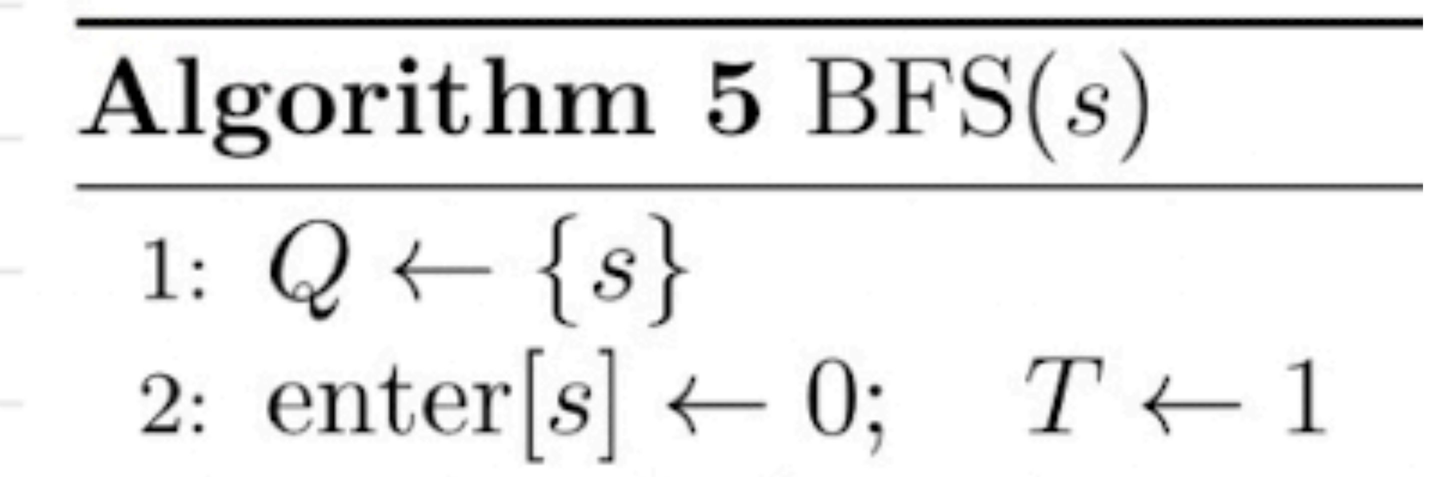
A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	

distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

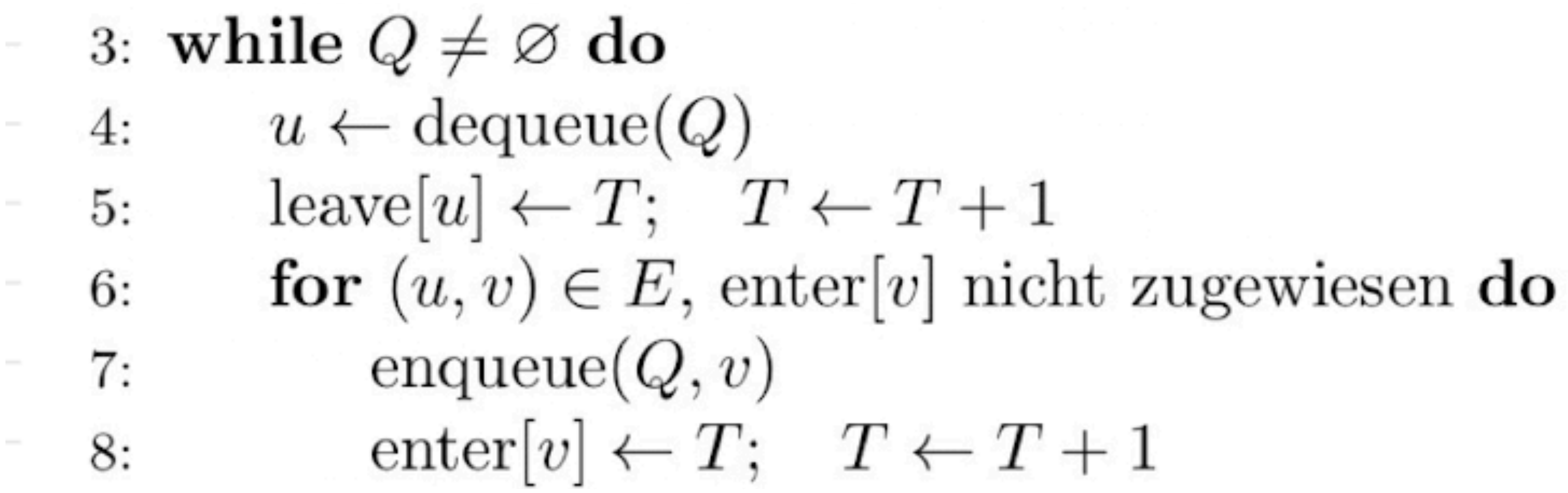
Graph Searchs

BFS - Example



Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter$[s] \leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
```



```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave$[u] \leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter$[v]$ nicht zugewiesen do
        enqueue($Q, v$)
        enter$[v] \leftarrow T$;    $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

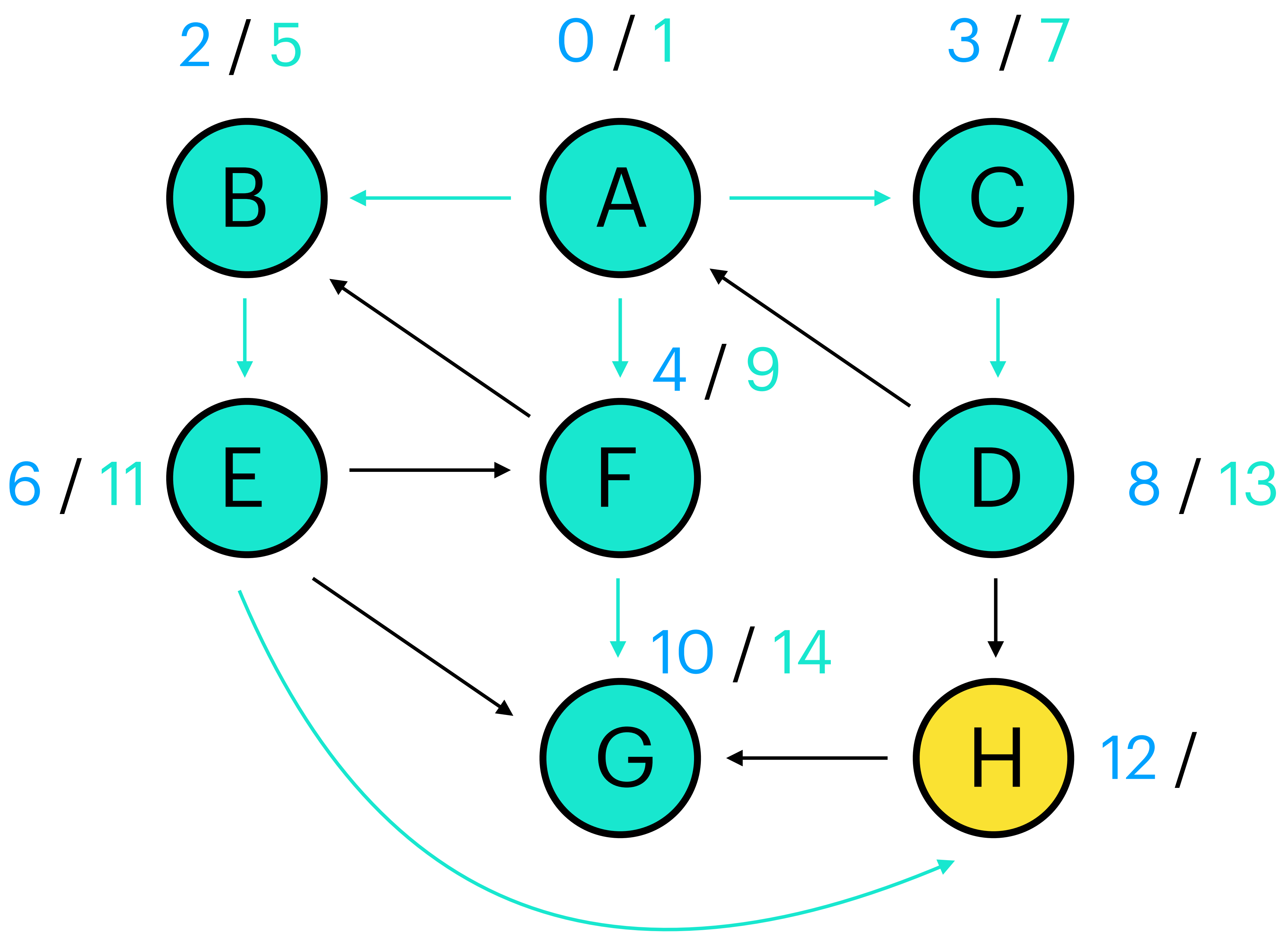
Q : H

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

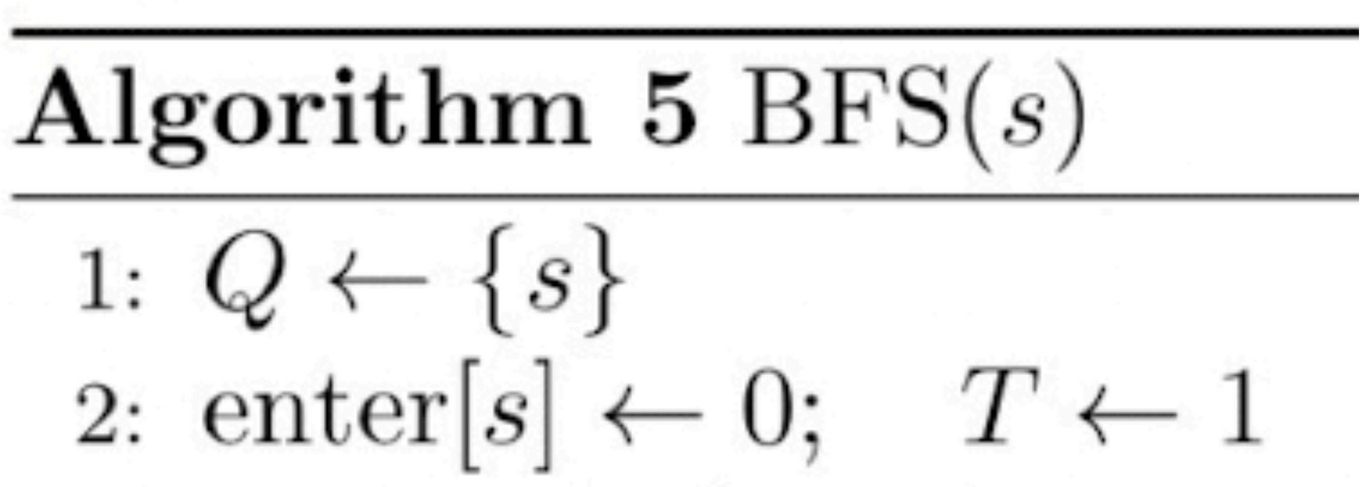
leave[] :

A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	



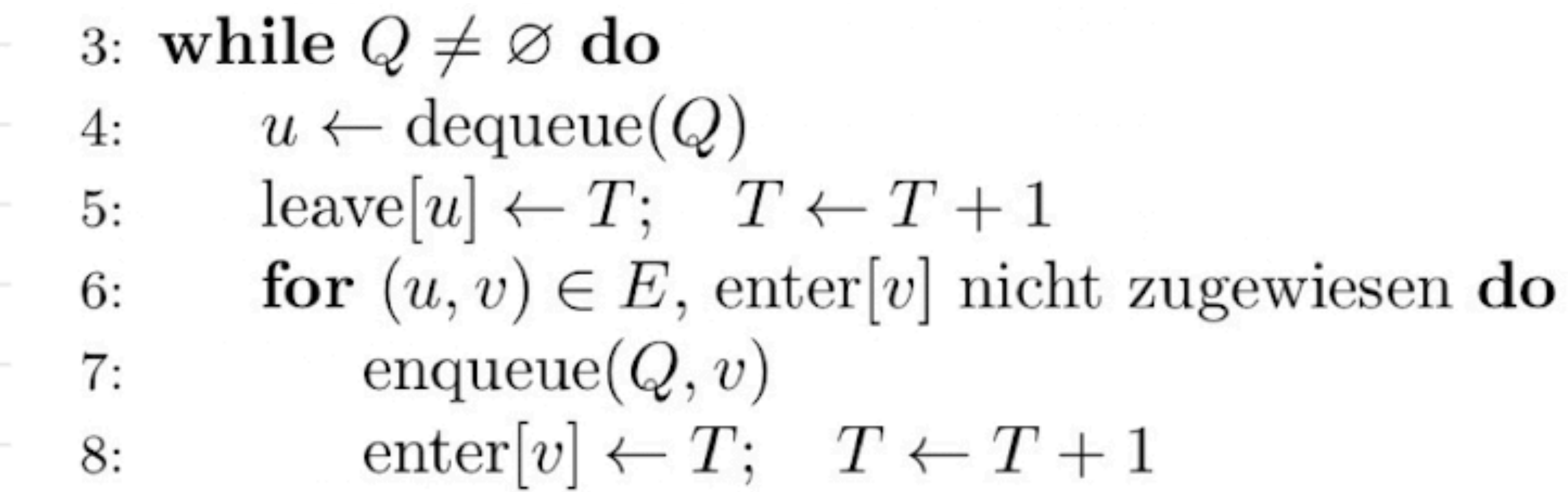
Graph Searchs

BFS - Example



```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
```

distance[s] = 0 ;

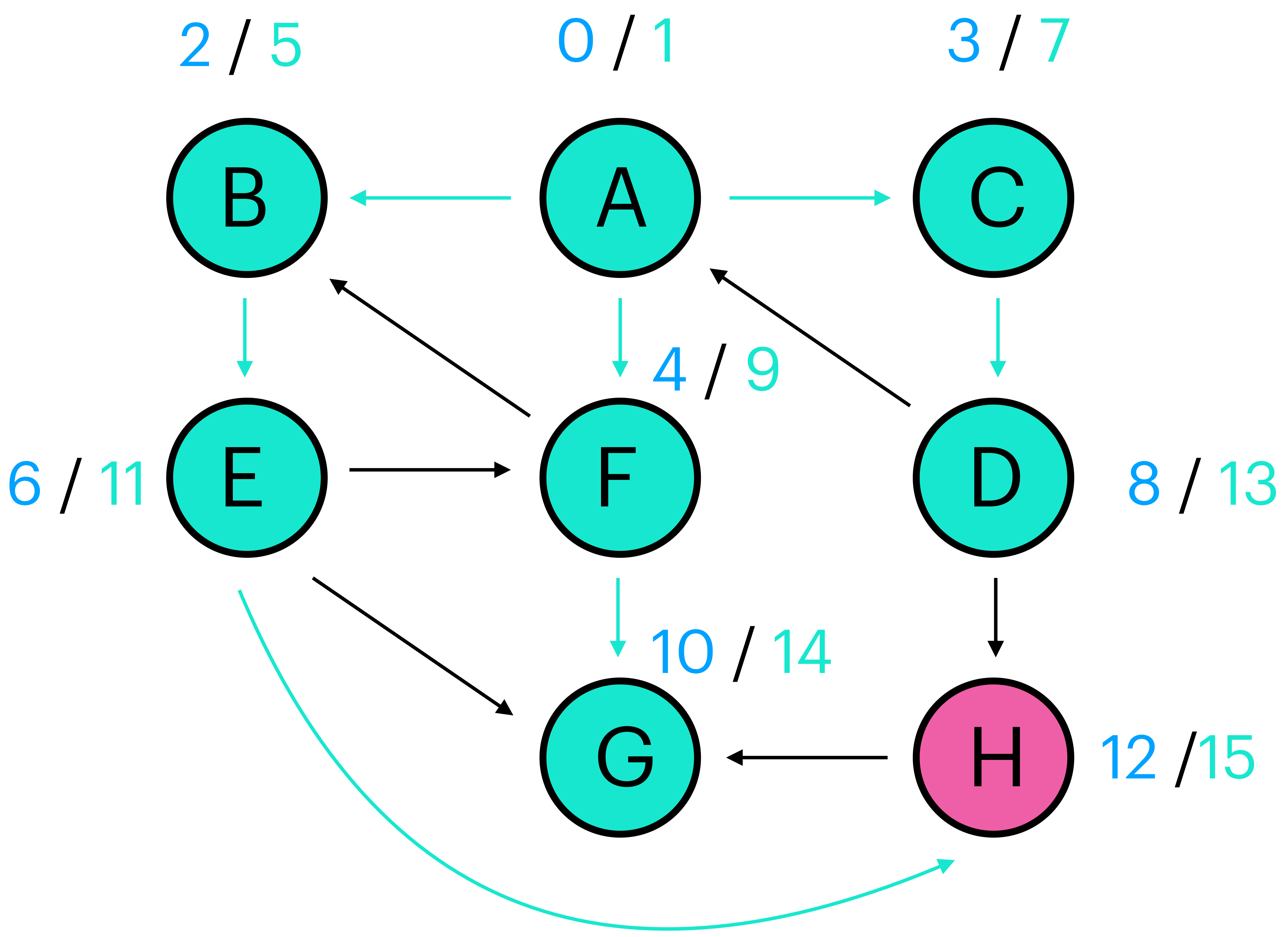


```
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
```

distance[v] <- distance[u] + 1 ;

Q :

u = H



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

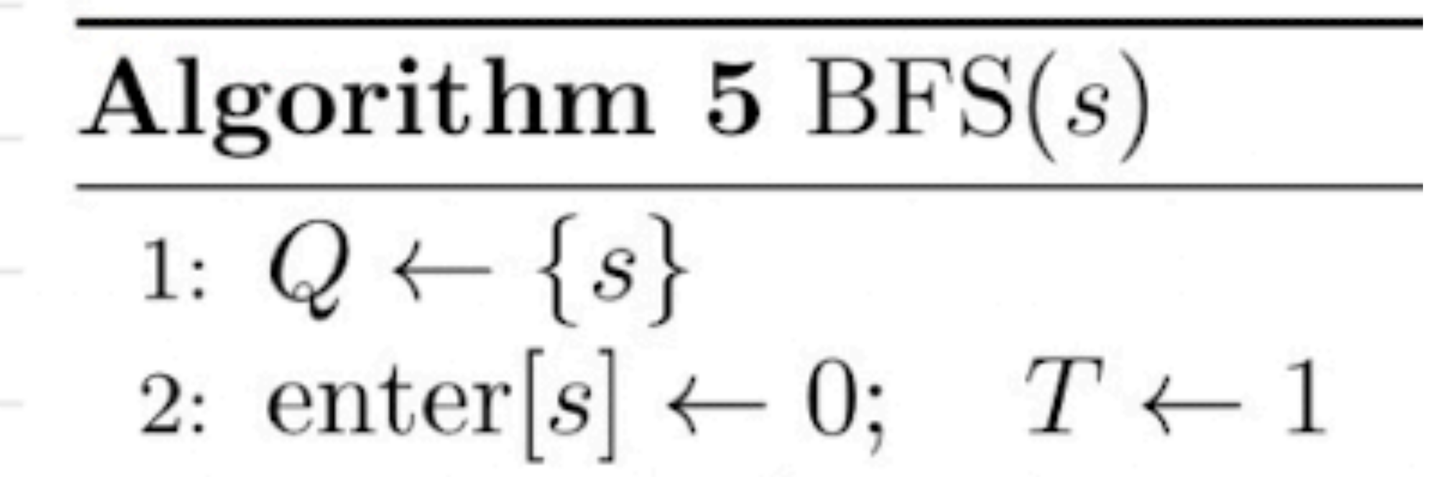
A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	15

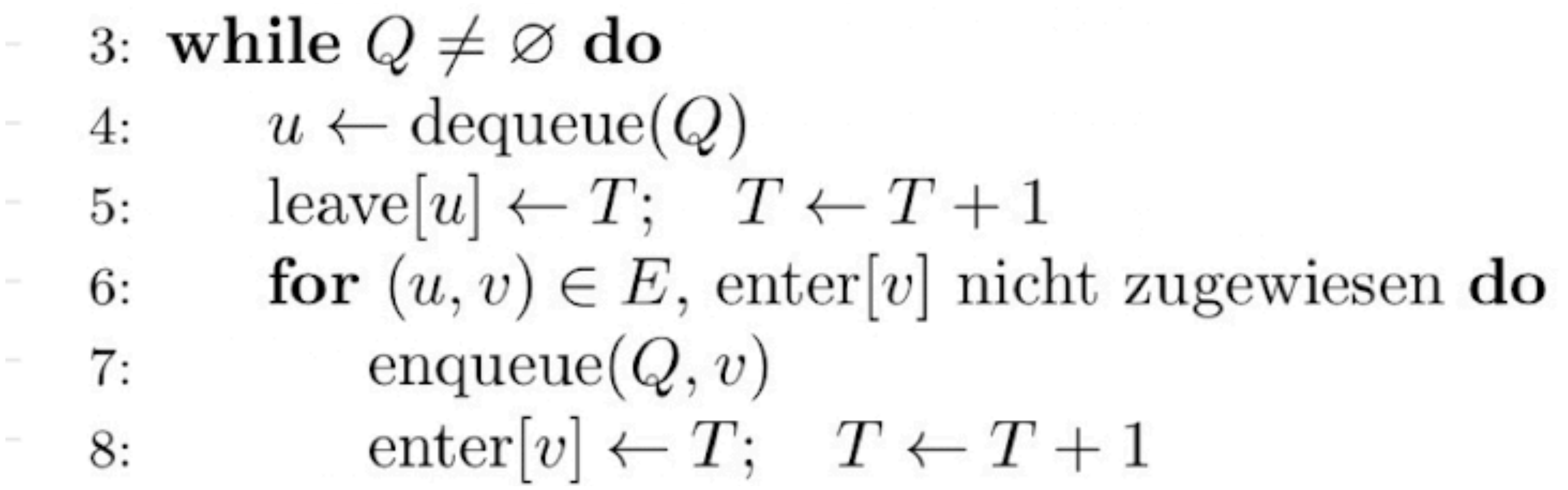
distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

Graph Searchs

BFS - Example





```
Algorithm 5 BFS($s$)
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;    $T \leftarrow 1$
distance[s] = 0 ;
while $Q \neq \varnothing$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;    $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;    $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q :

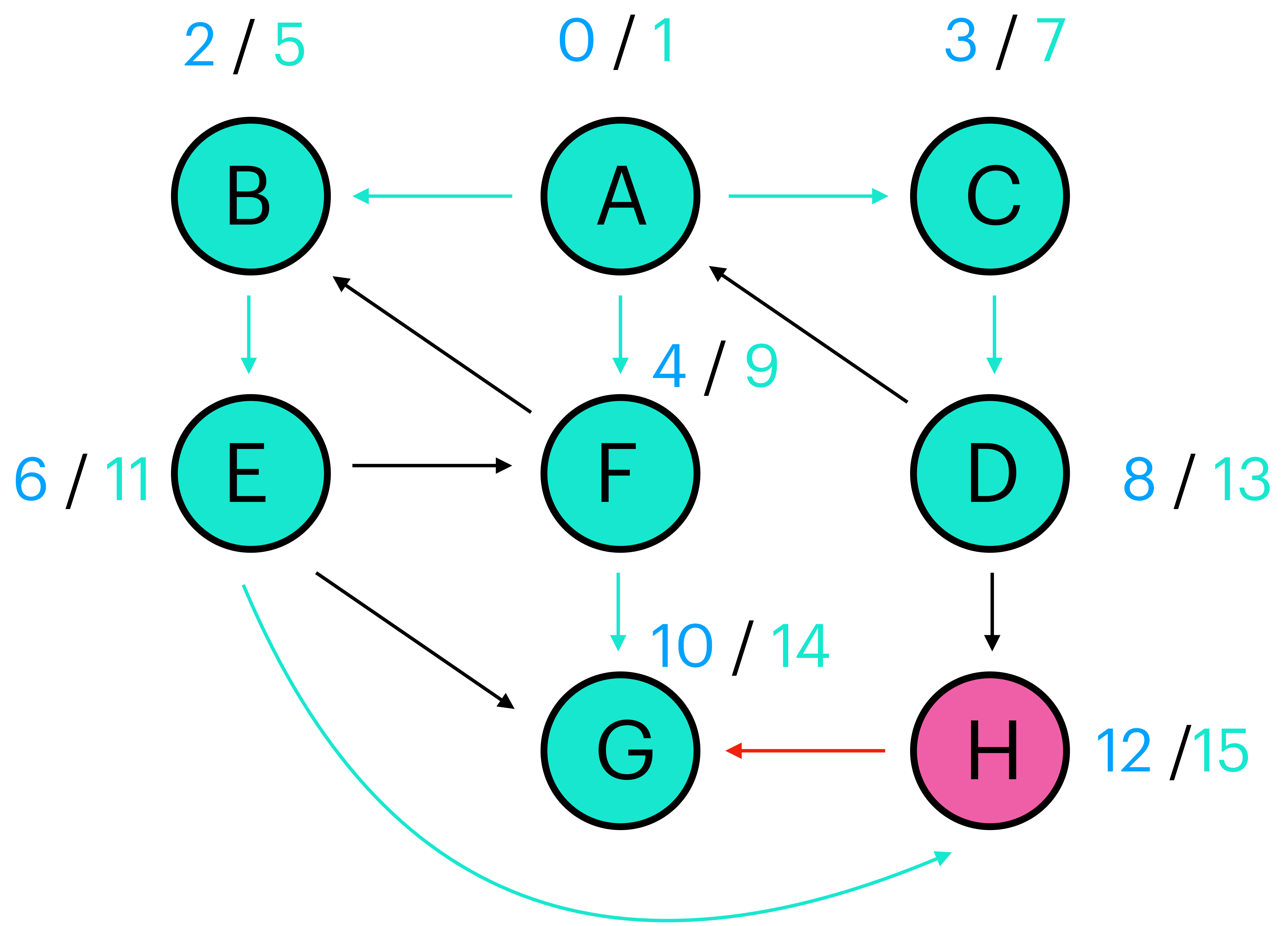
u = H

enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	15

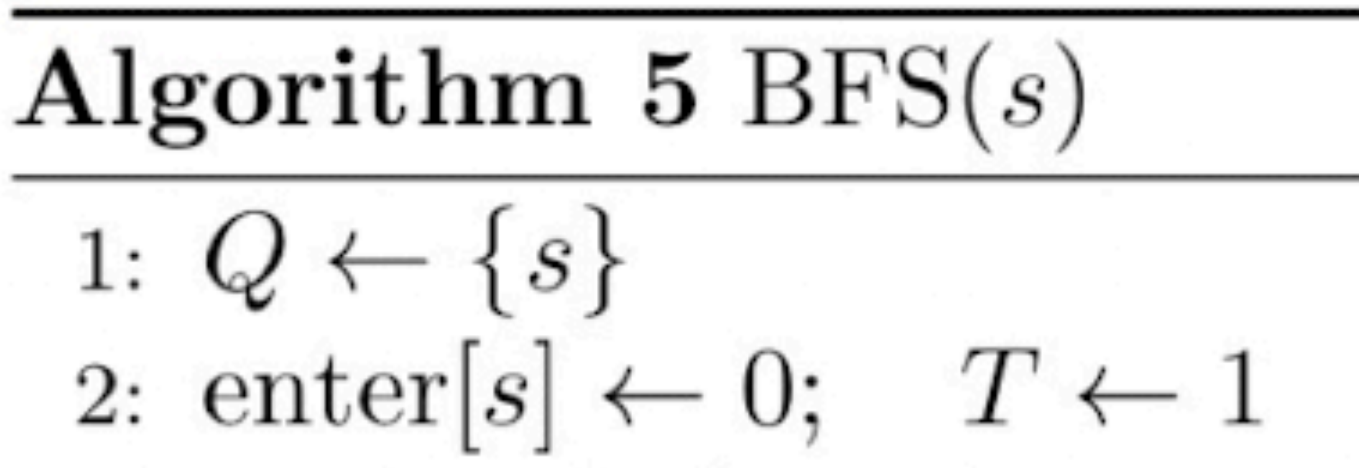


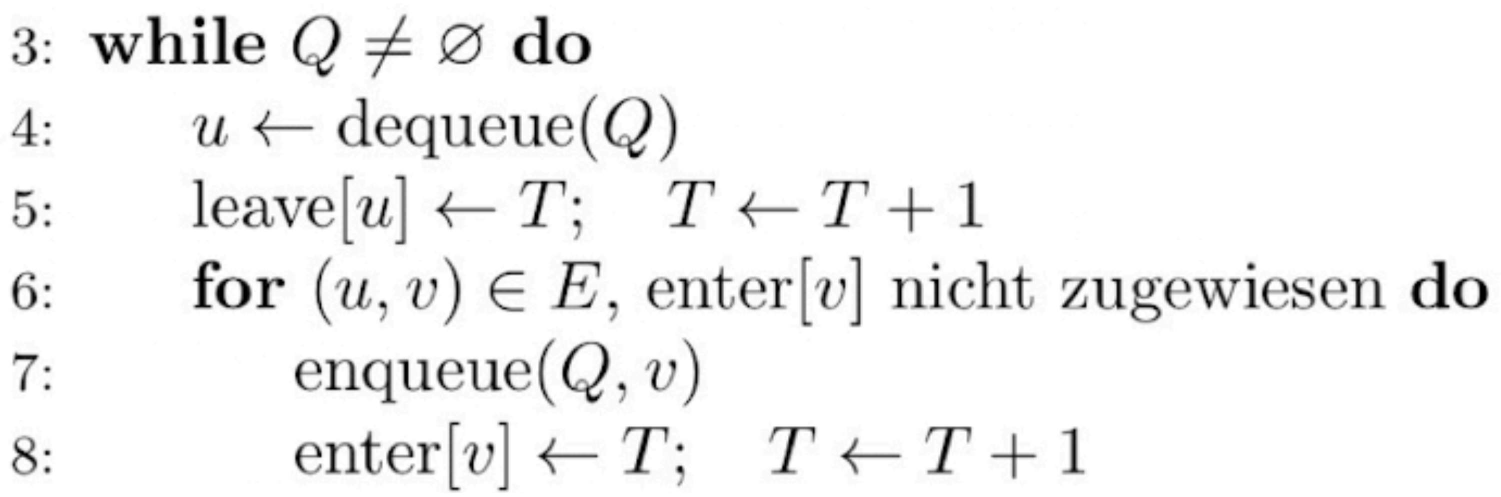
distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

Graph Searchs

BFS - Example



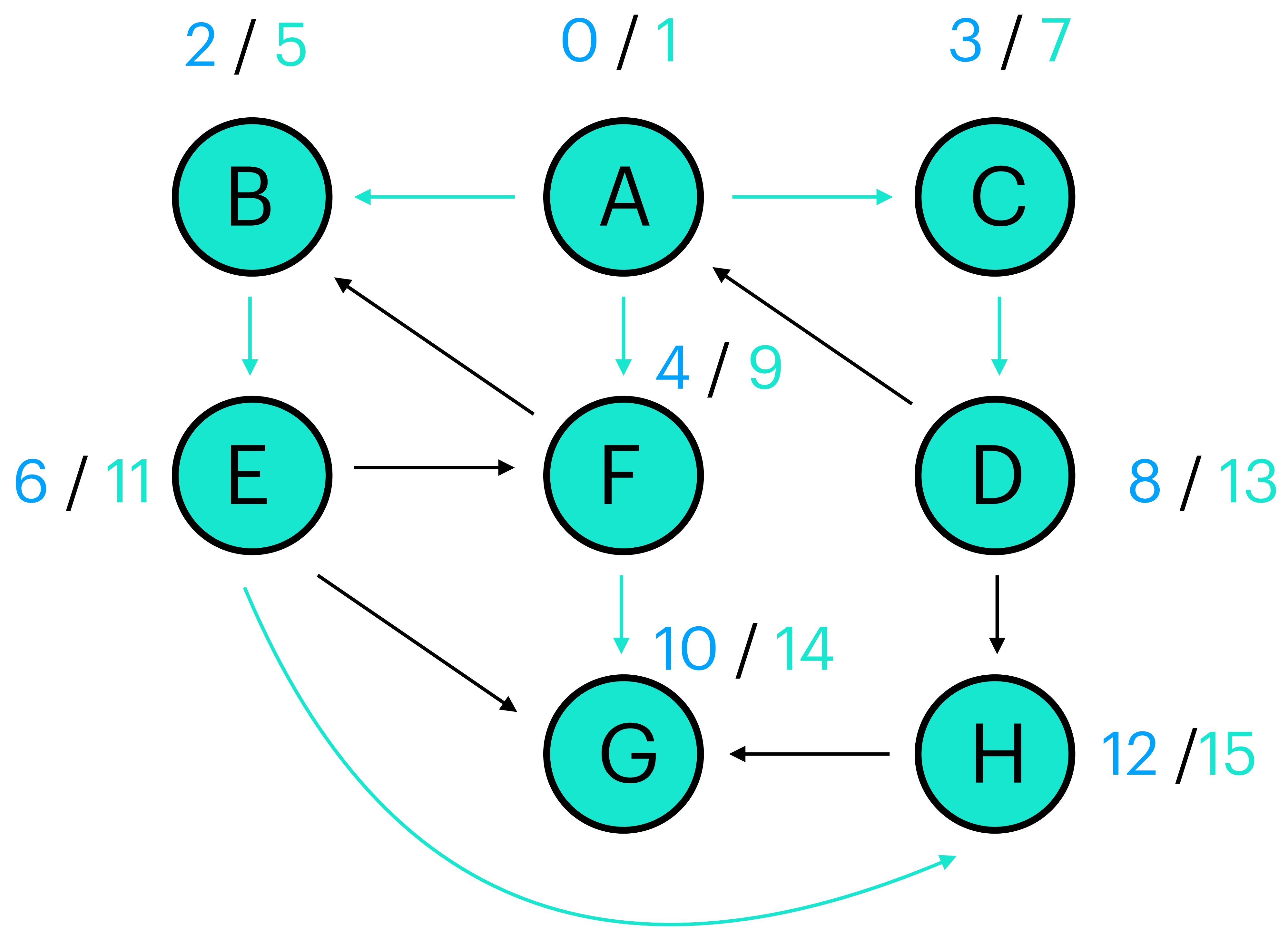


Algorithm 5 BFS(s)

```
$Q \leftarrow \{s\}$
enter[$s$] $\leftarrow 0$;   $T \leftarrow 1$
    distance[s] = 0 ;
while $Q \neq \emptyset$ do
    $u \leftarrow$ dequeue($Q$)
    leave[$u$] $\leftarrow T$;   $T \leftarrow T + 1$
    for $(u, v) \in E$, enter[$v$] nicht zugewiesen do
        enqueue($Q, v$)
        enter[$v$] $\leftarrow T$;   $T \leftarrow T + 1$
        distance[v] <- distance[u] + 1 ;
```

Q :

Q = ∅



enter[] :

A	B	C	D	E	F	G	H
0	2	3	8	6	4	10	12

leave[] :

A	B	C	D	E	F	G	H
1	5	7	13	11	9	14	15

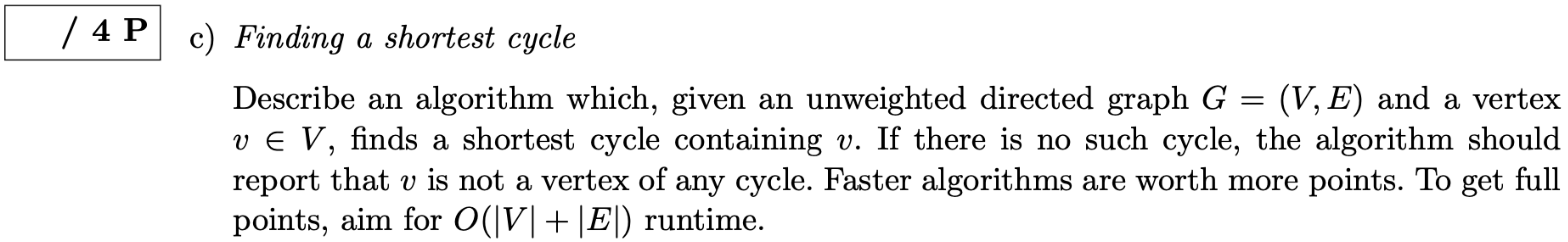
distance[] :

A	B	C	D	E	F	G	H
0	1	1	2	2	1	2	3

BFS

Exam Question

HS21



/ 4 P

c) *Finding a shortest cycle*

Describe an algorithm which, given an unweighted directed graph $G = (V, E)$ and a vertex $v \in V$, finds a shortest cycle containing v. If there is no such cycle, the algorithm should report that v is not a vertex of any cycle. Faster algorithms are worth more points. To get full points, aim for $O(|V| + |E|)$ runtime.

DFS, BFS

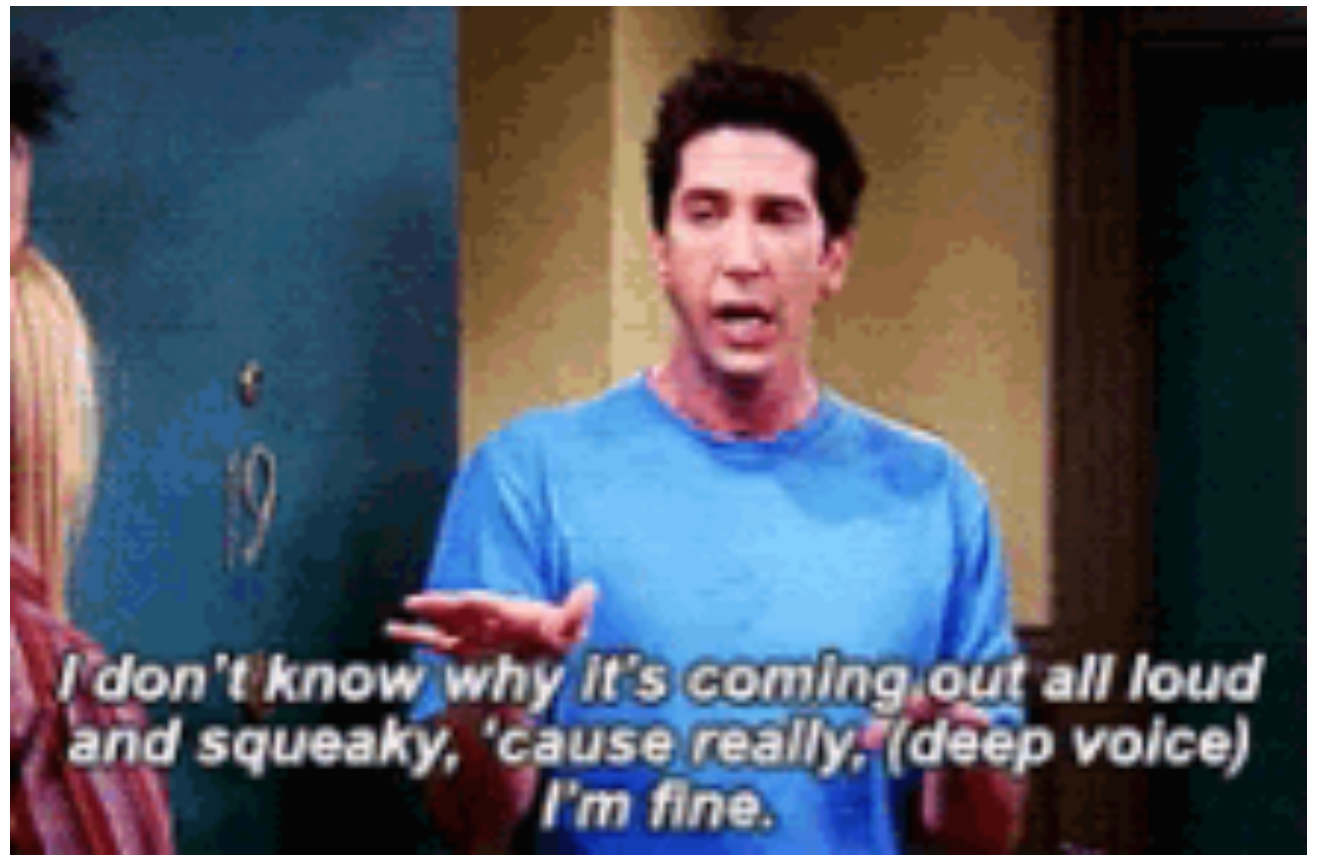
Exam Tipps

- DFS for **reachability** , finding components
- BFS for **distance** in unweighted Graph (Shortest Path in unweighted Graph)

- Theory
  - Usual new algorithm tipps
  - Watch out for the runtime !! O(|V| + |E|)
- Coding
  - Know how to implement both !! How ? ...

Let’s take a break

Code Expert - Graph Sets


I don't know why it's coming out all loud
and squeaky, 'cause really, (deep voice)
I'm fine.

Next Week ...

Shortest Path algorithms (one-to-all) ...

Questions

Feedbacks , Recommendations

Nil Ozer