

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ACS Web Labs Web

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:53:49

```
main.js
1 class Stack {
2     constructor() {
3         this.stack = [];
4     }
5
6     push(v) {
7         this.stack.push(v);
8     }
9
10    pop() {
11        return this.stack.pop();
12    }
13
14    empty() {
15        return this.stack.length === 0;
16    }
17}
```

And I'm going to add an empty()

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ACS Web Labs Web

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:53:29

```
main.js
1 class Stack {
2     constructor() {
3         this.stack = [];
4     }
5
6     push(v) {
7         this.stack.push(v);
8     }
9
10    pop() {
11        return this.stack.pop();
12    }
13
14    empty() {
15        if (this.stack.length === 0) {
16            return true;
17        } else {
18            return false;
19        }
20    }
21}
```

And for empty, I'm going to return

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:53:00

```
main.js
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Java: A `java.util.Stack` with `.push()`, `.pop()`, and `.empty()`.

Examples

```
adj_list: [
    [1],
    [2, 3],
    [4],
    []
]
start: 0
result: [0, 1, 2, 4, 3]
```

We're going to initialize the visited

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:52:04

```
main.js
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Java: A `java.util.Stack` with `.push()`, `.pop()`, and `.empty()`.

Examples

```
adj_list: [
    [1],
    [2, 3],
    [4],
    []
]
start: 0
result: [0, 1, 2, 4, 3]
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

BACK TO QUESTIONS Question 1 of 3 8:759:52:04

main.js

```
* Java: A java.util.Stack with .push(), .pop(), and .empty().
```

Examples

```
adj_list: [
  [1],
  [2,3],
  [4],
  [],
  [6]
]

(0) -> (1) -> (3)
|
(4) -<- (2)

start: 0
result: [0, 1, 2, 4, 3]
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

BACK TO QUESTIONS Question 1 of 3 8:759:48:59

main.js

```
* Java: A java.util.Stack with .push(), .pop(), and .empty().
```

Examples

```
adj_list: [
  [1],
  [2,3],
  [4],
  [],
  [6]
]

(0) -> (1) -> (3)
|
(4) -<- (2)

start: 0
result: [0, 1, 2, 4, 3]
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8:759:48:35

main.js

```
* Java: a java.util.Stack with .push() , .pop() , and .empty() .
```

Examples

```
adj_list: [
  [1],
  [2, 3],
  [4],
  [1],
  [6]
]

(0) -> (1) -> (3)
|           |
(4) -<- (2)

start: 0

result: [0, 1, 2, 4, 3]
```

So I'll call this n and let's just

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8:759:48:10

main.js

```
* Java: a java.util.Stack with .push() , .pop() , and .empty() .
```

Examples

```
adj_list: [
  [1],
  [2, 3],
  [4],
  [1],
  [6]
]

(0) -> (1) -> (3)
|           |
(4) -<- (2)

start: 0

result: [0, 1, 2, 4, 3]
```

Let's see, if n is in visited, we've

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:47:33

main.js

```
* Java: a java.util.Stack with .push(), .pop(), and .empty().

Examples

adj_list: [
    [1],
    [2, 3],
    [4],
    [1],
    [6]
]

(0) -> (1) -> (3)
|
| v
(4) <- (2)

start: 0

result: [0, 1, 2, 4, 3]
```

Code editor interface showing Java code for iterative DFS implementation.

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:47:10

main.js

```
constructor() {
    this.stack = [];
}

push(v) {
    this.stack.push(v);
}

pop() {
    return this.stack.pop();
}

empty() {
    return this.stack.length === 0;
}

function solution(adj_list, start) {
    // Initialize result array to empty
```

Code editor interface showing Java code for iterative DFS implementation.

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:46:44

```
main.js
37 // If it is visited, skip it
38 if (n in visited)
39     continue
40
41 // Mark as visited
42 visited[n] = true;
43
44 // Add to results
45 result.push(n);
46
47 // Go through all the neighbors
48 const neighbors = adj_list[n];
49
50 // Push each neighbor on the stack
51 for (let neighbor of neighbors)
52     |
53
54
55
56
57 }
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Web ★ ACS Web ★ Labs Web ★ Log Out

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:46:17

```
main.js
40 // If it is visited, skip it
41 if (n in visited)
42     continue
43
44 // Mark as visited
45 visited[n] = true;
46
47 // Add to results
48 result.push(n);
49
50 // Go through all the neighbors
51 const neighbors = adj_list[n];
52
53 // Push each neighbor on the stack
54 for (let neighbor of neighbors)
55     to_visit.push(neighbor);
56
57
58
59
60 }
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:45:55

main.js

```
* Java: A java.util.Stack with .push(), .pop(), and .empty().  
Examples  
adj_list: [  
    [1],  
    [2, 3],  
    [4],  
    [1],  
    [6]  
]  
  
(0) -> (1) -> (3)  
|  
(4) -<- (2)  
  
start: 0  
  
result: [0, 1, 2, 4, 3]
```

main.js

```
29 // Add the start node to the stack  
30 to_visit.push(start);  
31  
32 // While the stack isn't empty  
33 while (!to_visit.empty()) {  
34     // Get the next node to visit  
35     const n = to_visit.pop();  
36  
37     // If it is visited, skip it  
38     if (n in visited)  
39         continue;  
40  
41     // Mark as visited  
42     visited[n] = true;  
43  
44     // Add to results  
45     result.push(n);  
46  
47     // Go through all the neighbors
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

Question 1 of 3 8,759:45:55

main.js

```
* Java: A java.util.Stack with .push(), .pop(), and .empty().  
Examples  
adj_list: [  
    [1],  
    [2, 3],  
    [4],  
    [1],  
    [6]  
]  
  
(0) -> (1) -> (3)  
|  
(4) -<- (2)  
  
start: 0  
  
result: [0, 1, 2, 4, 3]
```

main.js

```
29 // Add the start node to the stack  
30 to_visit.push(start);  
31  
32 // While the stack isn't empty  
33 while (!to_visit.empty()) {  
34     // Get the next node to visit  
35     const n = to_visit.pop();  
36  
37     // If it is visited, skip it  
38     if (n in visited)  
39         continue;  
40  
41     // Mark as visited  
42     visited[n] = true;  
43  
44     // Add to results  
45     result.push(n);  
46  
47     // Go through all the neighbors
```

portal.bloomtech.com/student/course-materials

BloomTech Portal

Home Classes Course Materials Gradebook Sprint Challenges GCA Book Meeting Support

Iterative Depth-First Search Implementation

BACK TO QUESTIONS Question 1 of 3 8:759:45:34

main.js

```
* Java: a java.util.Stack with .push(), .pop(), and .empty().
```

adjList: [{1}, {2, 3}, {4}, {1}, {4}]
(0) -> (1) -> (3)
|
(4) -> (2)
atStart: 0
result: [0, 1, 2, 4, 3]

JavaScript

```
39     continue
40
41     // Mark as visited
42     visited[n] = true;
43
44     // Add to results
45     result.push(n);
46
47     // Go through all the neighbors
48     const neighbors = adjList[n];
49
50     // Push each neighbor on the stack
51     for (let neighbor of neighbors)
52       toVisit.push(neighbor);
53   }
54
55   // Return the result
56   return result;
57 }
```