

Stack and recursive method

There are several situations when recursive methods are quite handy

- DFS
- Traversing a binary search tree
- Looking for an item in a linked list

All the recursive algorithms can be transformed into a simple method with stacks

Depth-first search

Recursion

```
public void dfs(Vertex vertex) {  
  
    vertex.setVisited(true);  
    printf(vertex);  
  
    for(Vertex v : vertex.neighbours() ){  
        if( !v.isVisited() ){  
            dfs(v);  
        }  
    }  
}
```

Iterative approach with stack

```
public void dfs(Vertex vertex) {  
  
    Stack stack;  
    stack.push(vertex);  
  
    while( !stack.isEmpty() ){  
  
        actual = stack.pop();  
  
        for(Vertex v : actual .neighbours() ){  
            if( !v.isVisited() ){  
                v.setVisited(true);  
                stack.push(v);  
            }  
        }  
    }  
}
```

Fatorial

Recursion

```
public void factorial(int n) {  
  
    if( n == 0 )  
        return 1;
```

```
return n * factorial(n-1);  
}
```

What does it all have to do with stacks? The recursive function calls are pushed

onto the stack until we bump into the base case

- We keep backtracking: we know the base case so we know the subsolutions
- If there are too many function calls to be pushed onto the stack: the stack may get full ... no more space left
- Stack overflow !!!

Bruna Santos - March 30, 2018 09:58 am