

PSAT Week 8: Graph Algorithms II

Objectives: basic data structures, stack, queue, breadth-first search

Data Structures:

- A data structure describes a method for storing/organizing data/information
- Data structures typically support operations for adding and retrieving information

Stack:

- A **stack** is a data structure to store a collection of elements
- *push*: add an element to the stack
- *pop*: remove the element from the stack that was pushed last (and return it)
- Visualization: Stack of plates
- push to the top, pop from the top
- LIFO (last in first out)

Queue:

- A **queue** is a data structure to store a collection of elements
- *enqueue*: add an element to the queue
- *dequeue*: remove the earliest enqueued element from the queue (and return it)
- Visualization: Queue of people
- enqueue at the back, dequeue from the front
- FIFO (first in first out)

Task 1 (10 min):

- Perform the following operations for an initially empty stack:
push('E') push('A') push('S') pop() push('Y') pop() push('Q') push('U')
push('E') pop() pop() pop() push('S') push('T') pop() pop() pop()
push('I') push('O') pop() push('N') pop() pop() pop()
- Perform the following operations for an initially empty queue:
enq('E') enq('A') enq('S') deq() enq('Y') deq() enq('Q') enq('U') enq('E')
deq() deq() deq() enq('S') enq('T') deq() deq() deq() enq('I') enq('O')
deq() enq('N') deq() deq() deq()

Depth-First-Search algorithm (for starting node s and target node t):

push s to the stack

Repeat the following instructions until the stack is empty:

pop x from the stack

If x is not marked yet, perform the following instructions:

Mark x as 'visited'

If x is the target node t, report that the target has been found

Otherwise, scan through the edges that have x as one of their endpoints and perform push y for each such edge, where y denotes the other endpoint of the edge

Task 2 (20 min):

In the DFS algorithm we used a stack. Now consider the same algorithm with the stack replaced by a queue. In which order are the nodes of the graph visited in the resulting algorithm?

Breadth-First-Search algorithm (for starting node s and target node t):

enqueue s

Repeat the following instructions until the queue is empty:

 dequeue x

 If x is not marked yet, perform the following instructions:

 Mark x as 'visited'

 If x is the target node t, report that the target has been found

 Otherwise, scan through the edges that have x as one of their endpoints and perform enqueue y for each such edge, where y denotes the other endpoint of the edge

Distance Labels:

- We can extend the BFS algorithm to additionally compute a distance label for every node.
- $\text{dist}(s) = 0$ (when s is enqueued)
- $\text{dist}(y) = \text{dist}(x) + 1$ (when y is enqueued)
- This computes the distance from s to every node x, i.e., the length of the shortest path from s to x.