5118006-03 Data Structures

Stack

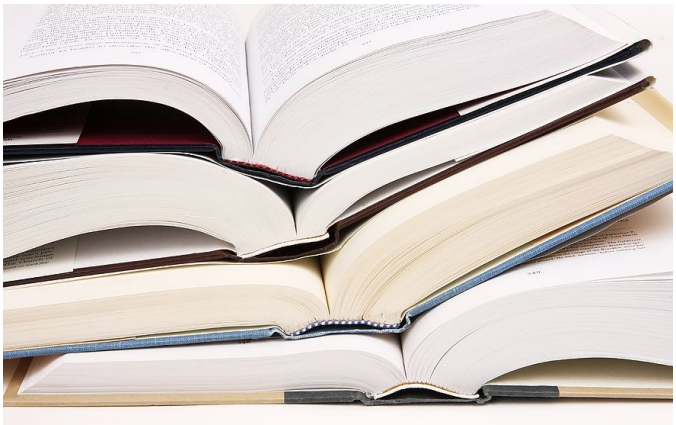
20 Mar 2024

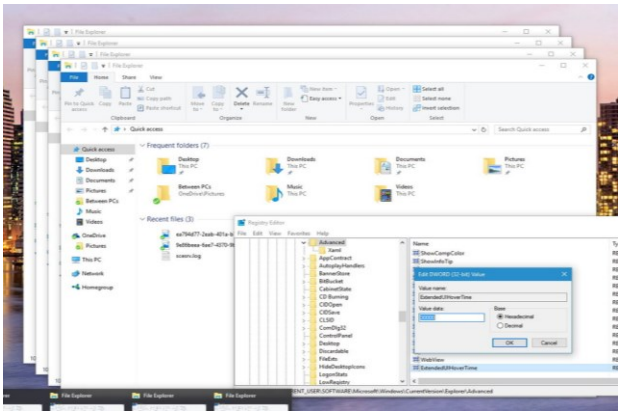
Shin Hong

Stack

- A stack is an ordered list where insertion and deletion is made only at one end
  - stack is also called LIFO (Last-In-First-Out)
  - the end to which an item is inserted/deleted is called top

- A stack is useful for storing temporal states in recursive search



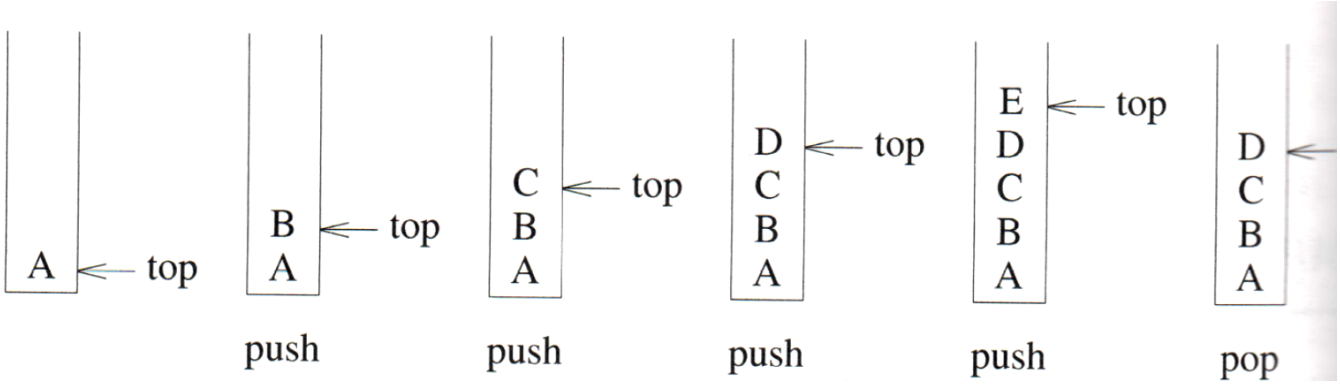


Abstract Data Type

- Data container
  - **buffer**: an array to hold elements
  - **capacity**: the capacity of the buffer array
  - **top**: an index of the array to place a next element if the buffer is not full, or the capacity of the buffer

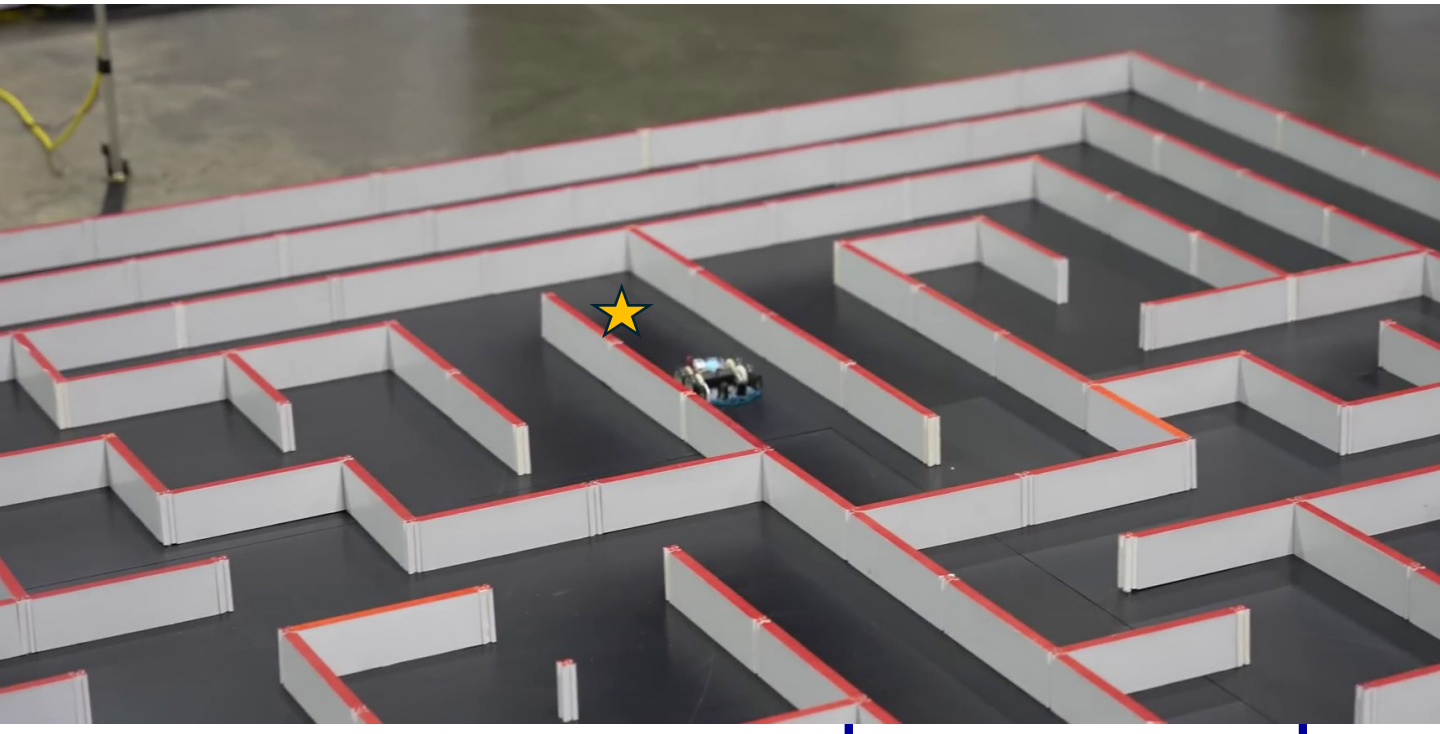
- Operations
  - **push(e)**: insert a new element **e** to the stack if the stack is not full
  - **pop()**: return the most recently inserted element if the stack is not empty
  - **isEmpty()** : return whether the stack has at least one element or not
  - **isFull()** : return whether the stack is full or not

Example


A
top
B
A
top
push
C
B
A
top
push
D
C
B
A
top
push
E
D
C
B
A
top
push
D
C
B
A
pop

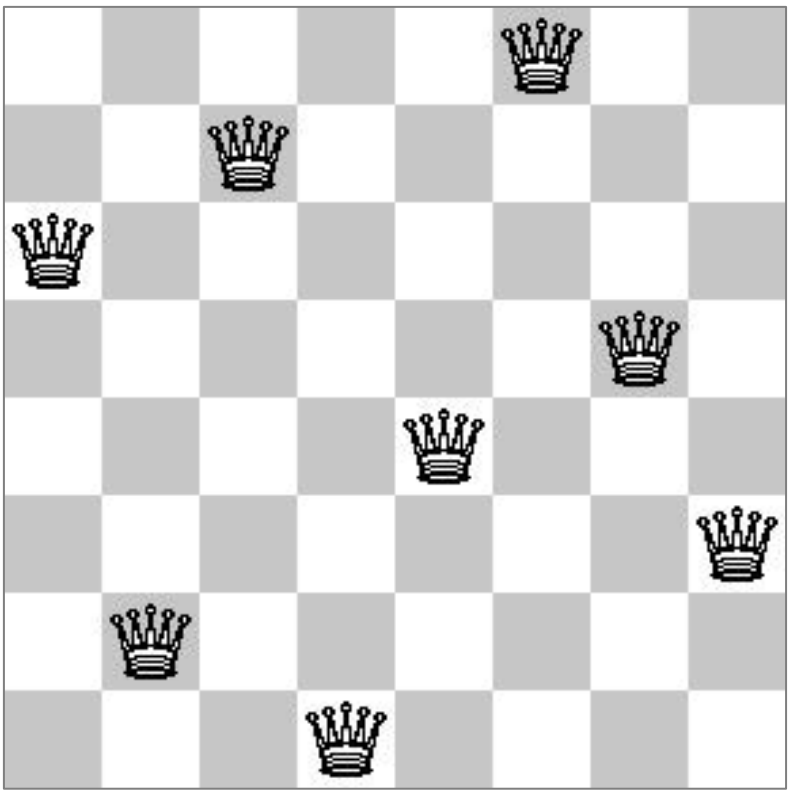
Case 1. Maze

- Find a path that consists of vertical and/or horizontal lines from the top-left corner (entrance) to the bottom-right corner
  - a player can move **up**, **down**, **left** or **right** to an empty cell
- Store the current path in a stack
  - each element represents the exploration status at a cell



Case 2. N-Queens Problem

- Find a placement of N queens on a checkboard such that they do not conflict with each other
  - Two queens cannot stand together if they are on the same vertical / horizontal / diagonal line



Case 3. Evaluating Expression

- An expression is a value, or one or more expressions connected with an operator

- Different notation to represent an arithmetic expression
  - Postfix: an operator is placed after its operands
  - Prefix: an operator is placed before its operands
  - Infix: a binary operator is placed between two operands
    - ambiguity

- Example
  - Postfix: `3 6 + 2 4 - * 7 +`
  - Prefix: `+ * + 3 6 - 2 4 7`
  - Infix : `((3 + 6) * (2 - 4)) + 7`

Implementation

- https://github.com/hongshin/DataStructures/tree/main/code/Mar20