

Design problems

Describe how you could solve each of the following problems using data structures we've learned so far. Specify the ADTs (List, Iterator, Stack, Queue) you would use, how they would be used to support the required operations, and how they should be implemented to optimize efficiency. Although lists can do anything stacks or queues can, give the most specific correct answer: if the functionality of a stack or queue is enough, pick that. Only answer "list" if neither a stack nor a queue suffices.

1. You are hired to help design software to help with a key airline operation: processing drink orders on a flight. Once the pilot gives the ok, airline staff walk from the back of the plane to the front taking everyone's order. When they reach the front, they begin preparing those orders. Drinks are prepared in groups of 10, that being the number of cups that can fit onto a carrying tray. The first group whose drinks are prepared are those in the front of the plane (first class!). The second group is the next 10 people further back in the plane and so on. You are in charge of developing a new application that will let airline staff take orders and then display those orders 10 at a time in the appropriate order.
2. You are in charge of making the lines at airport security run smoothly. Every day thousands of people pass through security. There are 3 main types of people who go through this line: economy class passengers, VIP passengers, and flight staff. There is one line but some people have the ability to cut through. The system should allow VIPs and flight staff go straight to the scanners. They are rare occurrences and only appear 1 in every 100 people.
3. The program is a simulation of an vertical stack in an ultimate frisbee game.¹ There's seven players and five of them are in a vertical line (cutters). The other two (handlers) are positioned horizontally in the back of the line (resembling the shape of an upside-down T). One of the handlers has the disk (frisbee) and is going to throw it to one of the cutters. The first cutter in the line tries to get it, and if they don't, they go back to the back of the line. This keeps happening until one gets the disk. That cutter then dumps it to the handler and goes to the back of the line, then it all starts again. The handler could also throw the disk to the other handler but they never join the line. The program should end after the line has reset 7 times. What ADT would you use to implement this simulation and how?
4. You're cleaning your dorm room, but every time you start one job you find another that needs your attention first! You try to put away your laundry, but it's all dirty! Then when you try to wash your laundry you find that your laundry basket is filled with junk you've been too lazy to put away, and when you try to put some stuff away you find some nasty old moldy food that you have to clean up, etc. Note that each time you discover a new task, it must be completed before the one that you were previously working on. What ADT would you use to help you keep track of all the subtasks you have to do before your room is finally clean?

¹This is a simplified description and not quite how the game is played.