Discussion questions

**Runtime**

1. The workload is conceptualized as the number of tasks required to complete a job. For searching through a box of animal crackers to see if it contains an elephant, the workload is determined by the number of crackers in the box (which would increase proportionally with increasing number of crackers).
   1. O(1)
   2. O(log n)
   3. O(n)
   4. O(n log n)
   5. O(n2)
   6. O(2n)

**Stacks & queues**

1. Pallets – stacks because you would want to get the most recent ones (the ones in the front) out first
2. Beer caps – a queue because you would want to put a cap on the first bottle you encounter
3. Equation – you take the parenthesis in order so most likely a queue

Two more situations to use queues would be a line in a bank (you would want to serve the customers in the order in which they lined up, first one in would be the first one served) and in baking batches of cookies (first cookies in the oven are the first ones out of the oven).

Two more situations for stacks include doing a stack of paperwork (first one on the pile gets attended to first) and grabbing a CD from a stack of CDs (not that they exist anymore).

**Linked Lists**

1. **Given the linked list below, which are the nodes?**

Apple, Berry and Cherry are the nodes

**What is the data for each node?**

A string and an arrow to the following node

**Where is the head?**

The box containing the string “Apple”

**Where is the tail?**

This linked list does not have a tail specified.

1. The difference is that with doubly-linked lists, the arrows go both ways meaning that you can travel forward or backward to the adjacent node (where as for singly, you can only travel in one direction).
2. Because then we would not have to traverse the linked list until we got to an arrow that pointed to None (i.e, the end of the linked-list).