

Module 8 Prompt

Q1:

A priority queue has a special characteristic that elements dequeued or popped from this data store follow a specific order, such as ascending or descending. This data structure can be used for some applications that ingest random data and the most prioritized data is able to get from this data structure. For example, CPU resource scheduling can use priority queues that store awaiting processes, and prioritized processes are going to be run after the CPU is available.

Q2:

Since traditional queues follow the FIFO rule, there would be an implicit prioritization if the incoming data is following a specific order of prioritization. Otherwise, the queue there's no any prioritization.

Q3:

The priority queue can be effectively implemented with Min/Max heap data structure, which allows a constant time/cost to access the min/max element.

Q4:

If the order of increases/decreases, the size and layout of the tree will be affected. Since the order means the number of children that each node can have. So if the order increases, it would not cause any change in size and the layout, whereas the decrease of the order requires to reconstruct the tree and would make the tree narrower and deeper.

Q5:

If a Huffman tree takes a different way to evaluate tie-breakers, it will change the way to compress the data. In class if two nodes with a single character, the tie-breaker is to order them alphabetically (B vs D), if we use the opposite order, it would change the result of the compression: 110 vs 111. This could cause a wrong result after decompression.