Taarvni Ananya   Homework 4   4/8/25

1) The data structure that best suits the listed operations is a min-heap because the operations need to look at the schedule and make the appropriate decisions. Min heap's priority queue runs an efficient program to add future events and determine when to execute the next event seamlessly

a) the minheap algorithm can perform an insert operation with its O(log n) time complexity because it keeps the smallest nodes near the root, ensuring a smoother and faster insertion with min heap when compared to other data structures

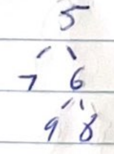
b) as for extracting the smallest time-stamp, min heap once again provides the best solution. A min heap features the smallest node at the root, and all children are larger than their parent. Because the root has the smallest value, it's easier with a minheap for this operation to find the smallest timestamp

Using heapq is the best solution for the min-heap

```
import heapq
Min_queue = []
a -> heapq.heappush(min_queue, (3, "words"))
b -> smallest = heapq.heappop(min_queue)
```

2) Bill's statement is wrong because the preorder traversal sequence is root → left → right. The times when root < left/right is with a min heap but this does not always guarantee that left < right. The root could also be equal to either values

A legit min-heap that proves Bill wrong is:

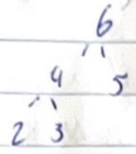


Here, 5 < 7 > 6, not 5 < 7 < 6. Left is larger than right and cannot be a valid preorder traversal. The min heap on the other hand is valid and legit

5 → 7 → 6 → 9 → 8 is not in nondecreasing order

3. Hillary's claim is wrong because the post order traversal sequence is left → right → root where left < right < root. Hillary claims that values need to be in decreasing order but this is not guaranteed as both min & max heaps have parent ≤ children and parent ≥ children respectively.

The following tree is legit and a valid max heap



Here, $6 > 4$, $6 > 8$, $4 > 2$, and $4 > 3$. This disrupts the postorder travel to be 2, 3, 4, 5, 6 where values are increasing, not decreasing