2.
To implement this I would use a binary heap. Every node would indicate how many items it had in the list. So if there needed to be something added to the one with the least amount in it, it would automatically go to the root. Each node represents a tour guide, the number represents how many items are in the list. If you were to add a visitor to the group with the smallest amount of people the time cost would depend. If by adding the visitor, the group would still remain the smallest, it would take O(1). If the group was no longer the largest and it had to restore the heap order property, it would take O(log n).