

Homework 2

1. Show that $O(2^{2n})$ is not equal to $O(2^n)$. (Hint: you can use proof by contradiction.)
2. In class we wrote a function `PhysIndex(A, i)` that computed the physical index (the actual memory location) into a circular array for a given logical index `i` (the position within the rotating buffer). Write a function `PhysIndexReverse(A, i)` that computes the physical index when iterating through the array in reverse order (so that logical index 0 corresponds to the tail rather than the head of the array).
3. Suppose you have a `ListNode N` with a `next` pointer and a `data` element. Each node contains integers. Write a *recursive* function that finds the sum of all elements in the list.
4. Suppose that the numbers $0, 1, \dots, 9$ are pushed onto a stack in that order, but pops occur at random points between various pushes (multiple pops can occur in a row and every item is popped in the end). The following is a valid sequence in which the values in the stack could have been popped:
3, 2, 6, 5, 7, 4, 1, 0, 9, 8
Explain clearly why
3, 2, 6, 4, 7, 5, 1, 0, 9, 8
is not a valid sequence in which the values could have been popped off the stack.
5. Suppose that an initially empty stack S has executed a total of 25 push operations, 12 top (peek) operations, and 10 pop operations, 3 of which raised Empty errors that were caught and ignored. Determine the current size of S .
6. Implement a function `transfer(S, T)` that transfers all of the elements from S to T , with the top element of S being the first added to T and the bottom element of S being the new top element of T .
7. Show how to use the `transfer` function and two temporary stacks to replace the contents of stack S with the same elements, but in reverse order.
8. Suppose that an initially empty queue Q is represented by a circular array of capacity 10 and the first item is to be inserted at physical index 0. 8 items are then enqueued, 5 are dequeued, and then 6 more items are enqueued. Find the physical indices of the first and last elements of the queue.
9. Write a *recursive* function that removes all elements from a queue.