Question 2

Obtain an explicit formula for the following recurrence using one of the techniques seen in class.

I begin by substituting a few (k number of) times the T(n-1) component to look for a pattern:

$$k = 0;\ T(n) = 2T(n-1) + 4^n + 1$$
$$k = 1;\ T(n) = 2(2T(n-2) + 4^{n-1} + 1) + 4^n + 1$$
$$k = 1;\ T(n) = 4T(n-2) + 2(4^{n-1}) + 2 + 4^n + 1$$
$$k = 2;\ T(n) = 4(2T(n-3) + 4^{n-2} + 1) + 2(4^{n-1}) + 2 + 4^n + 1$$
$$k = 2;\ T(n) = 8T(n-3) + 4(4^{n-2}) + 4 + 2(4^{n-1}) + 2 + 4^n + 1$$
$$k = 3;\ T(n) = 8(2T(n-4) + 4^{n-3} + 1) + 4(4^{n-2}) + 4 + 2(4^{n-1}) + 2 + 4^n + 1$$
$$k = 3;\ T(n) = 16T(n-4) + 8(4^{n-3}) + 8 + 4(4^{n-2}) + 4 + 2(4^{n-1}) + 2 + 4^n + 1$$

A pattern is emerging that the formula after k such substitutions would be:

$$T(n) = 2^{k+1}T(n-k-1) + \sum_{i=0}^{k} 2^i + \sum_{i=0}^{k} 2^i \left(4^{n-i}\right)$$

Given that we know recursion stops when T(0)=0, we can eliminate the recursive portion of the formula by setting k=n-1, and since the recursive portion becomes zero at n=0, we can also eliminate that component of the formula.

$$T(n) = \sum_{i=0}^{n-1} 2^i + \sum_{i=0}^{n} 2^i \left(4^{n-i}\right)$$

The $\sum_{i=0}^{n} 2^i \left(4^{n-i}\right)$ component can be rearranged (the work was done in rough) to become:

$$\sum_{i=0}^{n} 2^i \left(4^{n-i}\right) = \sum_{i=0}^{2n} 2^i - \sum_{i=0}^{n} 2^i$$

Thus the new formula is:

$$T(n) = \sum_{i=0}^{n-1} 2^i + \sum_{i=0}^{2n} 2^i - \sum_{i=0}^{n} 2^i$$
$$T(n) = \frac{1-2^n}{1-2} + \frac{1-2^{(2n+1)}}{1-2} - \frac{1-2^{(n+1)}}{1-2}$$
$$T(n) = -\left(1 - 2^n + 1 - 2^{(2n+1)} - 1 - 2^{(n+1)}\right)$$
$$T(n) = -1 + 2^n - 1 + 2(4^n) + 1 - 2(2^n)$$
$$T(n) = 2(4^n) - 2^n - 1$$

Question 3

Write an algorithm that prints the appropriate initial ordering for any given number n of cards.

Algorithm: orderCards(n)
Input: int n
Output: Prints the correct card ordering.

```
int a[n]
a[n-1]=n
for i=1 to (n-1)
   temp=a[n-1]
   k=n-1
   while (k>(n-i)) do
      a[k]=a[k-1]
      k=k-1
   end while
   a[k]=temp
   a[k-1]=n-i
end for
for i=1 to n print a[i-1]
```

Question 4

Prove that $log(n!) \in \Theta(n\log(n))$

To prove that $log(n!) \in \Theta(n\log(n))$ I must show that $\log(n!) \in O(n\log(n))$ and $n\log(n) \in O(\log(n!))$

Given that $n!$ can be rewritten as $1 \times 2 \times \ldots \times n$ and $\log(ab) = \log(a) + \log(b)$, $\log(n!)$ can be rewritten as:

$$\log(n!) = \sum_{i=1}^{n} \log(i) = \log(1) + \log(2) + \cdots + \log(n)$$

Which will always be less than:

$$n\log(n) = \sum_{i=1}^{n} \log(n) = \log(n) + \cdots + \log(n)\,; \log(n)\, is\ repeated\ n\ times$$

Thus:

$$\log(n!) \leq n\log(n)$$

Therefore $\log(n!) \in O(n\log(n))$

If I remove the first half of the terms from the first summation with the factorial, I have

$$\log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2} + 1\right) + \log\left(\frac{n}{2} + 2\right) + \cdots + \log(n)\,; a\ total\ of\ k\ terms$$

Which is always less than $\log(n!)$, but always greater than:

$$\log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}\right) + \cdots + \log\left(\frac{n}{2}\right); \log\left(\frac{n}{2}\right) is\ repeated\ k\ times$$

I can rearrange the above to:

$$\log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}\right) + \cdots + \log\left(\frac{n}{2}\right) = \frac{n}{2}\log\left(\frac{n}{2}\right) = \frac{n}{2}(\log_2(n) - \log_2 2) = \frac{n}{2}\log(n) - \frac{n}{2}$$

Thus for a sufficiently small constant c

$$\log(n!) \geq cn\log(n)$$

Therefore $\log(n!) \in \Omega(n\log(n))$