Homework 2 Solutions

1. Algorithm Enigma

   a) The algorithm returns "True" if its input matrix is symmetric and "False" if it is not

   b) Comparison of two matrix elements

   c) $C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1-i) = \frac{n(n-1)}{2}$

   d)The algorithm's efficiency class is O(n^2)because it compares each element of the upper triangular part of the matrix in the worst case.

2. The algorithm alternates walking right and left, exponentially increasing the distance from the starting point. For each iteration i, it moves 2^i steps to the right, returns to the start, then moves 2^i steps to the left, and returns again.

   If the distance to door is n, where $2^{k-1} < n <= 2^k$, , total steps required are estimated by

$$\sum_{i=0}^{k-1} 4.2^i + 3.2^k < 14\,n$$

   Thus the algorithm runs in O(n).

3. The number of different ways to climb an n-stair staircase if each step is either 1 or 2 stairs can be represented by the recurrence relation:
   W(n)=W(n−1)+W(n−2) for n ≥3 with base cases W(1)=1and W(2)= 2.
   • W(n) = F(n+1) for n>=1

4. To prove - $c_1$nlgn <= lg(n!) <= c_2.n.lgn

   lg(n!) = lg(1x2x3…n) while n lg n = lg(n^n) = lg(n x n… x n) for c_1 = 1 and n0 = 2, lg(n!) <= $c_1$nlgn. —Eq 1

   Consider the first half of lg(n!) i.e. lg(n/2 x .. n-1 x n) >= n/2 * lg(n/2)
   therefore n/2 * lg(n/2) <= lg(n!)
   (n/2) lg(n) - (n/2) <=lg(n!)

Compared to (n/2)logn, n/2 gets negligible
$c_2 = 1/2$ and $n > 2$

$(c_2)(n)\lg(n) <= \lg(n!)$ —Eq 2

From Eq 1 and Eq2 === $\lg (n!) = \Theta (n \lg n)$.

5. The number n can be represented in binary using k bits if $2^{k-1} <= n < 2^k$, taking the logarithm base 2 of both sides gives $k-1 <= \log_2 n < k$, so $k = \log_2 n + 1$.