Midterm Solutions

1. O(log n) < O(n) < O(n^2) < O(10^n) < O(n!) < O(n^n)

2.

To merge sorted lists $A = a_1, \ldots, a_n$ and $B = b_1, \ldots, b_n$:
  Maintain a *Current* pointer into each list, initialized to
    point to the front elements
  While both lists are nonempty:
    Let a_i and b_j be the elements pointed to by the *Current* pointer
    Append the smaller of these two to the output list
    Advance the *Current* pointer in the list from which the
      smaller element was selected
  EndWhile
  Once one list is empty, append the remainder of the other list
      to the output

3a. Yes.

3b. Yes

4.

```
def climbStairs(cost):
        #We'll treat the "top floor" as a step to reach
        nSteps = len(cost) + 1
        minimum_cost = [0]*nSteps

        # Start iteration from step 2, since minimum cost of reaching
        # step 0 and step 1 is 0
        for i in range(2, nSteps):
                take_one_step = minimum_cost[i-1] + cost[i-1]
                take_two_steps = minimum_cost[i-2] + cost[i-2]
                minimum_cost[i] = min(take_one_step, take_two_steps)

        return minimum_cost[nSteps-1];
```

5. False. False. True. True. False

6. Same as Homework 1, question 8.

7.

```
def minCost(costs):
        M[0][0] = costs[0][0];          #red
        M[0][1] = costs[0][1];          #green
        M[0][2] = costs[0][2];          #blue

        for i in range(1, n):
                M[i][0] = costs[i][0] + min(M[i-1][1], M[i-1][2])
                M[i][1] = costs[i][1] + min(M[i-1][0], M[i-1][2])
                M[i][2] = costs[i][2] + min(M[i-1][0], M[i-1][2])

        return min(M[n-1])
```