Homework 1

-Exercise 1

The algorithm Binary-Int-Sqrt is iterative. Modify it to make it a recursive algorithm.

*def* RecursiveSqrt(*low*, *high*, *n*):

    mid = math.floor((low + high) / 2)

    if mid \*\* 2 <= n and (mid \*\* 2) + 1 > n:

        return mid

    elif mid \*\* 2 <= n:

        return RecursiveSqrt(mid, high, n)

    else:

        return RecursiveSqrt(low, mid, n)

This is a recursive version of the binary square root algorithm. It uses the same general steps as the iterative version but calls the function instead of looping

-Exercise 2

There are n adults in town A and they all need to go to town B. There is only a single motorbike  
 available which is owned by two boys.  
 The motorbike can carry only one adult or up to two boys at a time (note that at one least person is needed to ride a bike). Using the motorbike, all the adults need to reach town B from town A. Show how this can be accomplished while, at the end, leaving the motorbike with the two boys in town A. Show the correctness of your algorithm by using mathematical induction and analyze the number of trips needed. (Hint: Use decrease and conquer to reduce the problem size.)

With 0 adults the problem is trivial, both boys and the bike are already in town A, 0 trips

With 1 adult in town A both boys would have to ride to town B first and drop 1 off since the bike can only hold 1 adult. Then one boy takes the bike back to A, the 1 adult in A rides the bike to B and the single boy in B rides the bike back to A resulting in both boys and the bike in A

With more than 1 adult this algorithm can simply be repeated n number of times where n is the number of adults that need to travel from A to B. The base case of 1 adult takes 4 trips to complete so with n adults the total number of trips needed would be 4n

-Exercise 3

Prove that For all using weak mathematical induction

Base case: n = 7 => 2187 < 5040 => True

Inductive hypothesis: Assume that For all k >= 7

< (k + 1)!

Inductive Hypothesis leaves us with which is always true with

So, the proof holds

-Exercise 4

Using strong mathematical induction, prove that every integer can be expressed in binary. That  
 is, show that, for any positive integer n, there exist integers a0, a1, ..., ak in {0, 1} such that

Note: k is used in the formula so I unlike class I will be using m instead

Base Case: P(1) => a0 = 1 so = 1 => so 1 can be represented in binary

Assume that P(2), P(3), … , P(m) are true

Inductive step:

P(m + 1) = P(m) + P(1), since P(1) has been shown in the base case to be representable in binary as, and P(m) is known to be true from the inductive hypothesis then P(m + 1) must also be true by induction.

-Exercise 5

Rank the following functions by order of growth. That is, find an arrangement f1, f2, ..., of the  
 functions satisfying f1 = O(f2), f2 = O(f3), and so on. Justify your ordering, i.e., show why  
 f1 = O(f2), f2 = O(f3), and so on.

is O()

3n is O(n)

n

is O()

Log7n

n!

32log n

(Log n) log n

From largest to smallest n!, , , n, 32log n, 3n, Log7n, , (Log n) log n,

-Exercise 6

Base case: n=0, P(0) = = 1

Inductive hypothesis: Assume that the sum is equal to for P(1), P(2), … P(k)

P(k + 1) = + => + => + => =>

=> if k+2 is some m then we have => which holds the proof by strong induction

This cannot be done via weak induction because as we discussed in class last week, to prove P(m) you must have also proven P(m - 1), and P(m – 2), … P(0) this is because of the summation term and means that strong induction must be used