1. Acknowledge all sources and collaborations. If you do not give an acknowledgement statement, your assignment may not be graded.

2. Find positive constants c_0 and c_1 such that, for all positive integers n,

   $c_0 n^3 < 27n^3 + 13n^2 + 873(\lg n)^3 < c_1 n^3$. Justify briefly.

3. Define $\alpha = (1+\sqrt{5})/2$. Define $f(0) = 0$, $f(1) = 1$, and $f(n) = f(n-1) + f(n-2)$ for all $n \geq 2$. Define $T(0) = T(1) = 4$ and $T(n) = T(n-1) + T(n-2) + 7$ for all $n \geq 2$.

   (i) Prove, for all integers $n \geq 3$, $f(n) > \alpha^{n-2}$.

   (ii) Prove, for all integers $n \geq 1$, $T(n) < 18f(n)$. Hint. use some results from the seminar (see version revised today).

4. (i) Show the output from the call `ff(4)`.

```
def ff(n):
  L = [1, 6]
  for j in range(2,n+1):    # j ranges from 2 to n
    L.append( L[j-2]+L[j-1] )
    # (*) invariant:  j is the index of the last element of L
  print L[n]
```

   (ii) Finish the proof of the claim.

   Claim: each time execution reaches `(*)`, the invariant holds.

   Proof. By induction on the variable `j`.

   Base case. In Python, list indices start at 0, so the first time execution reaches the for loop, L has its initial 2 elements, so the first time execution reaches line `(*)`, L has had exactly 1 element appended (its value is `1+6=7`), so L has exactly 3 elements, so the index of the last element is 2. Also, the first time execution reaches `(*)`, `j` is 2. So the invariant holds when `j` is 2.

   Inductive case. Let t be any integer ≥ 2. Assume that the invariant holds when execution reaches line `(*)` and `j=t`. We want to show that the invariant holds when execution reaches line `(*)` and `j=t+1`.

   So, assume execution reaches line `(*)` with `j=t+1`. Now ... (finish the proof) ...

5. (i) Trace the execution of the algorithm below with input `x=29` and `y=11`.

   (ii) Give the runtime as a function of k, assuming x and y each have k bits.

```
def mr(x,y): #
  if (x==0):
    return 0
  z = mr(x/2,y)
  if (1==x%2):
    return z + z + y
  return z + z
```