cmput 304 2023 study questions 2

1. Rearrange the following lines of code, and indent properly, so that `myroot(n)` computes square roots using the old-school square root algorithm.

```
L = basehundred(n)
def myroot(n):
digit += 1
digit = 0
for dd in L: #dd  next digit-pair from n
print(r)
print(r, '=', s+digit, '*', digit, '+ ',end='')
q, r = 0,0   # quotient, remainder
q, r = 10*q, 100*r + dd
q, r = q + digit, r - (s+digit)*digit
return q, r
s = 20*q   #LHS multiplier
while ((s+digit+1)*(digit+1) <= r):
```

2. Define $T(n)$ as the total number of calls to `fib( )` made by the initial call `fib(n)`.

```
def fib(n):
  if (n<=1):
    return n
  return fib(n-1) + fib(n-2)
```

a) Explain why $T(0) = T(1) = 1$.

b) Explain why, for $n \geq 2$, $T(n) = 1 + T(n-1) + T(n-2)$.

c) Prove by induction on n that $T(n) = 2f(n+1) - 1$, where $f(n)$ is Fibonacci(n).

d) Using c, prove that $T(n)$ is in $\Theta(g^n)$, where g is the golden ratio.

3. The Fibonacci number $f(n)$ is defined as 0 if n is 0, 1 if n is 1, and $f(n-1) + f(n-2)$ for all integers $n \geq 2$. Prove by induction on j that, for all non-negative integers j, the value of a after line 4 has executed exactly j times is $f(j)$.

```
def ifib(n):                #line 0
  a,b = 0,1                 #line 1
  for _ in range(n):        #line 2
    print(a)                #line 3
    a,b = b, a+b            #line 4
  return a                  #line 5
```

4. Recall from class that the every-case runtime of the usual addition algorithm is in $\Theta(n)$, where n is the size (number of bits) of the input. Show that the worstcase runtime of the usual multiplication algorithm is in $\Theta(n^2)$. Justify carefully.

5. Usually, we talk about complexity of algorithms. Here, we talk about the complexity of a problem. We say that the worstcase runtime complexity of *the addition problem* is in $\Theta(n)$, because A) there is an algorithm that runs in $\Theta(n)$ time, and B) no algorithm runs in $o(n)$ time. In your own words, prove B.

6. For the longest increasing subsequence problem, what is the optimal substructure property that allows the design of a dynamic programming algorithm that runs in worstcase polytime?

   For the following two problems, define $f(j)$ so that, for each index j, $f(j)$ is the length of a LIS ending at position j.

7. Trace the LIS algorithm from class on the following input. Explain in detail how you compute $f(0)$, $f(1)$ and $f(2)$.

j	0	1	2	3	4	5	6	7	8	9
S	11	2	9	6	3	12	0	4	8	5
f										

8. Let S be the sequence $(s_0, s_1, \ldots, s_9)$. Define $f(j)$ so that, for each index j, $f(j)$ is the length of a LIS ending at position j. Assume that $f(9) = 5$. Assume that $(s_1, s_3, s_4, s_6, s_9)$ is an increasing subsequence.

   For each $f(j)$ below, give the set $Z(j)$ of possible values of $f(j)$ consistent with the above information.

   a) Explain why $Z(0) = \{1\}$.

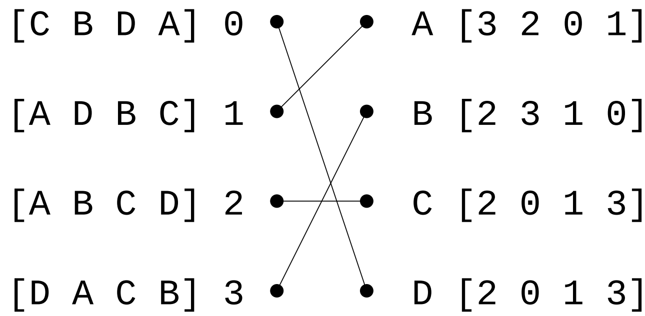
   b) Explain why $Z(1) = \{1\}$.

   c) Explain why $Z(2) = \{1, 2\}$.

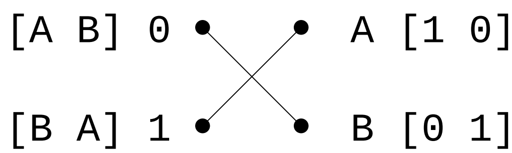
   d) For each remaining j, give $Z(j)$. You do not have to justify your answer.

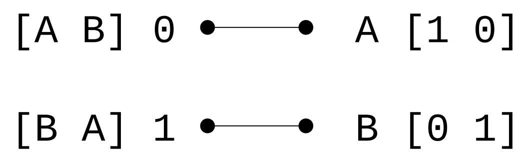
j	0	1	2	3	4	5	6	7	8	9
Z(j)	{1}	{1}	{1,2}	_____	_____	_____	_____	_____	_____	_____

9. For the matching below, for each unmatched pair of nodes (h, r) with h in $\{0, 1, 2, 3\}$ and r in $\{A, B, C, D\}$, determine whether $\{h, r\}$ is an unhappy couple. Justify each answer.

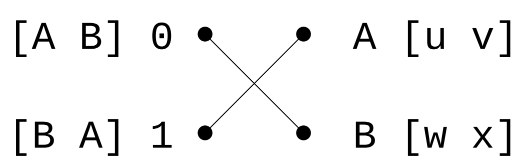


10. For each bipartite preference system and matching below, is the matching stable? Justify.





11. a) Give all possible assignments of values to `u,v,w,x` below so the bipartite system is valid and the matching is stable. Hint: u,v = 1,0 or 0,1, same for w,x.



b) Repeat a) so that the system is valid and the matching is unstable.