1) a) $\boxed{\text{Base case:}}$ We are given $f(0) = 0$ and since $0 \leq 0$ using the given definition, the base case satisfies.

$\boxed{\text{Assume } n = k \text{ holds:}}$ For this function $f: \mathbb{N} \to \mathbb{N}$ there are two outcomes, excluding the base case:
1. $f(n) = f(n-1)$
2. $f(n) = f(n-1) + 1$.

For 1, assume that $f(n) \leq n$, $f(k) \leq k$, and $f(k+1) \leq k+1$.

$\Rightarrow$ Substitute n with k: $f(k) = f(k-1)$.

$\Rightarrow$ Add 1: $f(k+1) = f(k)$

So then $f(k+1) \leq k$ because $f(k) \leq k$

So $f(k+1) \leq k+1$ since only the RHS is getting larger and the LHS stays the same. Thus $f(n) \leq n$ for all values of n.

$\boxed{\text{Assume } n = k+1 \text{ holds:}}$

~~Substitute $f(k) = f(k$~~

$\Rightarrow$ Substitute $f(n) = f(n-1) + 1$ with k:
$f(k) = f(k-1) + 1$.

$\Rightarrow$ Add 1: $f(k+1) = f(k) + 1$

We know that $f(k+1) = f(k) + 1$ because ~~we can again substitute $f(k)$ with $f(k+1)$ from 1.~~

~~This results in $f(k+1) \leq f(k+1) + 1$.~~

We need to prove $f(k+1) \leq k+1$ and we assume $f(k) \leq k$.

Adding one to both sides won't change the operation sign, so $f(k) + 1 \leq k+1$.

Given 2 we can substitute $f(k)+1$ with $f(k+1)$

So we have proven that given condition 2 $f(k)$ gives $f(k+1)$. Because base case satisfies, and conditions 1 and 2 work to construct next element, by PMI for any f and all $n \in \mathbb{N}$, $f(n) \leq n$.

1) b) Objective: prove by contradicting that there are countably many slowly increasing functions.

Bijective functions mean that ~~there are~~ a 1 to 1 mapping exists between $\mathbb{N}$ and the number of functions. Disproving that this exists for ~~the~~ the number of increasing functions is our proof.

Create a table such that it contains two axes - one with all the functions & $f_0 \to f_n$ and one with functions of f_n acting on each step from $1 \to n$.

For each element in the table, assign 1 or 0 if they belong to the set that contains slowly increasing functions.

For illustrative purposes, these 1's & 0's can be randomly assigned:

	$f_n(0)$	$f_n(1)$	$f_n(2)$	$\cdots$	$f_n(n)$
f_0	1	0	1	$\cdots$	0
f_1	1	1	0	$\cdots$	0
f_2	0	1	1	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
f_n	1	0	0		1

Flip the values $(0 \to 1,\ 1 \to 0)$ for elements on the diagonal to get a unique function $g: \mathbb{N} \to \mathbb{N}$ where g is defined:

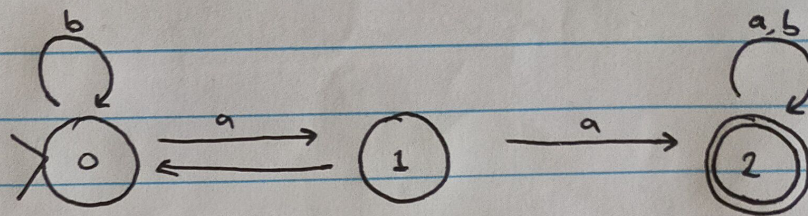
$$g \begin{cases} 0 & \text{if } f_k(k) = 1 \\ 1 & \text{if } f_k(k) = 0 \end{cases}$$

Function g is not already in the table going up to f_n, i.e. $g \neq f_k$. This is because flipping the value causes g to necessarily be unreachable by any function from ~~from~~ $f_0 \to f_n$. This means that there cannot be a bijective 1 to 1 mapping of $\mathbb{N}$ to the number of functions, since there will always be a remaining, & unique function g which cannot be mapped, causing there to be an uncountable number of slowly increasing functions, ~~[illegible]~~ as shown by proof from contradiction.

2) a) **CONTAINS-AA:**

Regular expression: $(a \cup b)^* aa (a \cup b)^*$

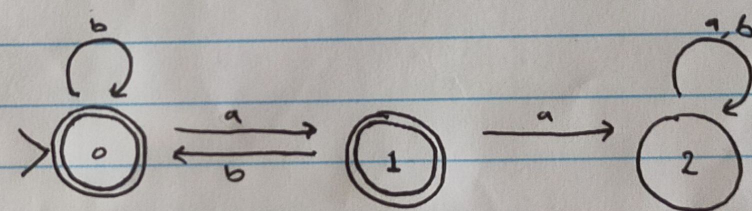
DFA:



b) **HAS-NO-AA:**

Regular expression: $(ab \cup b)^* a?$

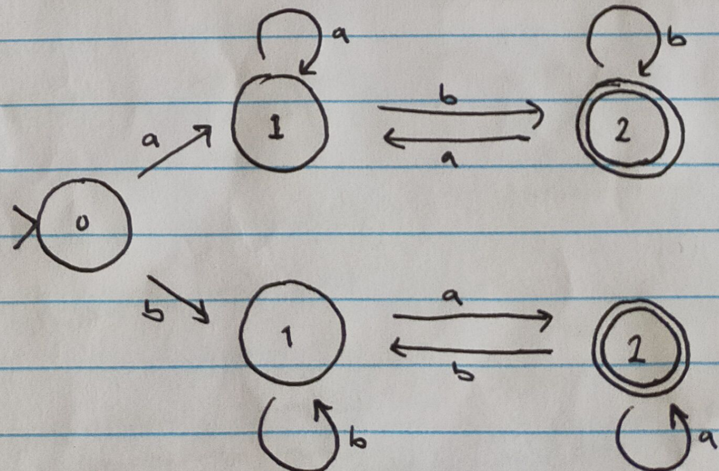
DFA:



c) **DIFFERENT-START-AND-FINISH:**

Regular expression: $(a(a \cup b)^* b) \cup (b(a \cup b)^* a)$

DFA:



3) a) Simple relationship: $f(n) = f(n-1) + f(n-2)$.

b) Proof by induction:

Base case: There is only one way to make a string of length zero, so the number of strings of length zero can be written as: $f(0) = 1$.

There are two letters, a and b. There are two ways to make a string of length 1. $f(1) = 2$.

length	number of strings	example
$f(0)$	1	" ".
$f(1)$	2.	"a", "b".
$f(2)$	3	"ab", "ba", "bb".

Check that $f(2) = 1 + 2$, since $f(n) = f(n-1) + f(n-2)$.
Indeed, checking by hand, there are 3 strings of length 2.

Assume $n=k$ holds: $f(k) = f(k-1) + f(k-2)$

There are two ways to make a valid string longer by one letter, without forming "aa".

⇒ Either: add a "b" to the end of the string. This ensures "aa" will not be present in the new string because no additional a was added to the already valid string.

⇒ Or: if the string ends in "b", add an "a" to the valid string. Again, "aa" will not be formed.

Consequently, new valid strings of a greater length are necessarily constructed using the strings before it.

Assume $n=k+1$ holds:

Given $f(k)=a$, and $f(k-1)=b$

$$f(k+1) = f(k) + f(k-1).$$
$$= a+1 = a + b.$$

~~Given $f(k)=a$, and $f(k-1)=b$~~

How it works:

f(1): a
b

f(2): ab
bb
ba

f(3): abb
bbb
bab
aba
bba

(2) No. strings in $f(3)$ is made from $f(1)$, since $f(1)$ had b's applied to end in step $f(2)$, meaning a's can be concatenated in $f(3)$

(1) Number of strings in $f(3)$ is made from $f(2)$ since b's can be added to each string of $f(2)$.

Therefore $f(3) = f(2) + f(1)$.