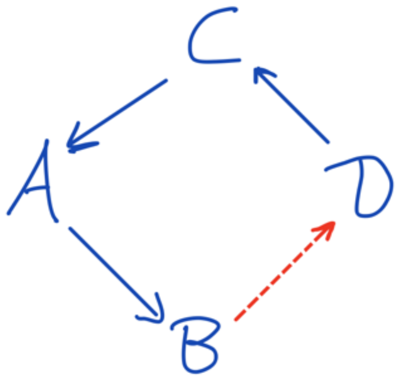
CS50X – 2020 – TIDEMAN.C

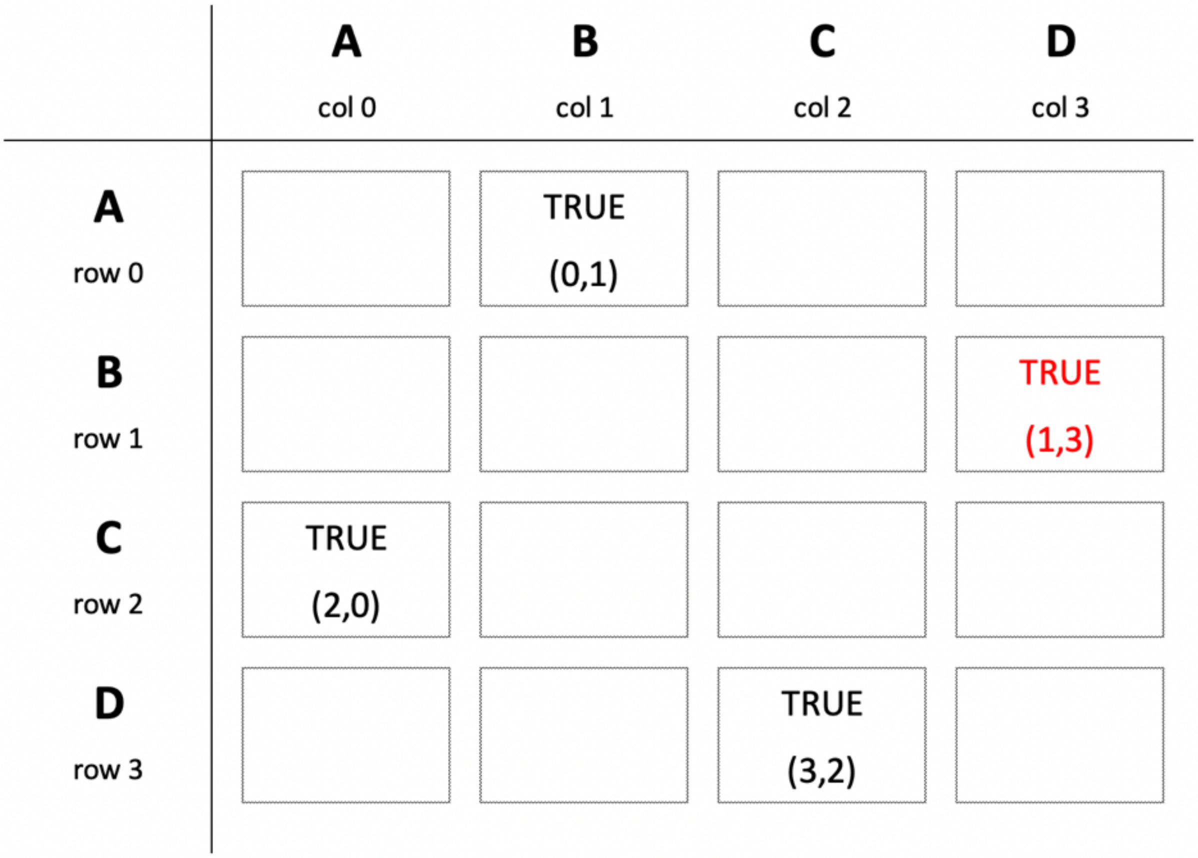
UNDERSTANDING THE LOCK_PAIRS FUNCTION

Shoutout and thanks to **robin#9650** on cs50 discord for the idea and concept.

Our locked_pairs graph may currently look like the image below. Now we want to add the edge from B–>D and check, wether or not this creates a cycle (which it obviously would).



Our current matrix looks like the following:

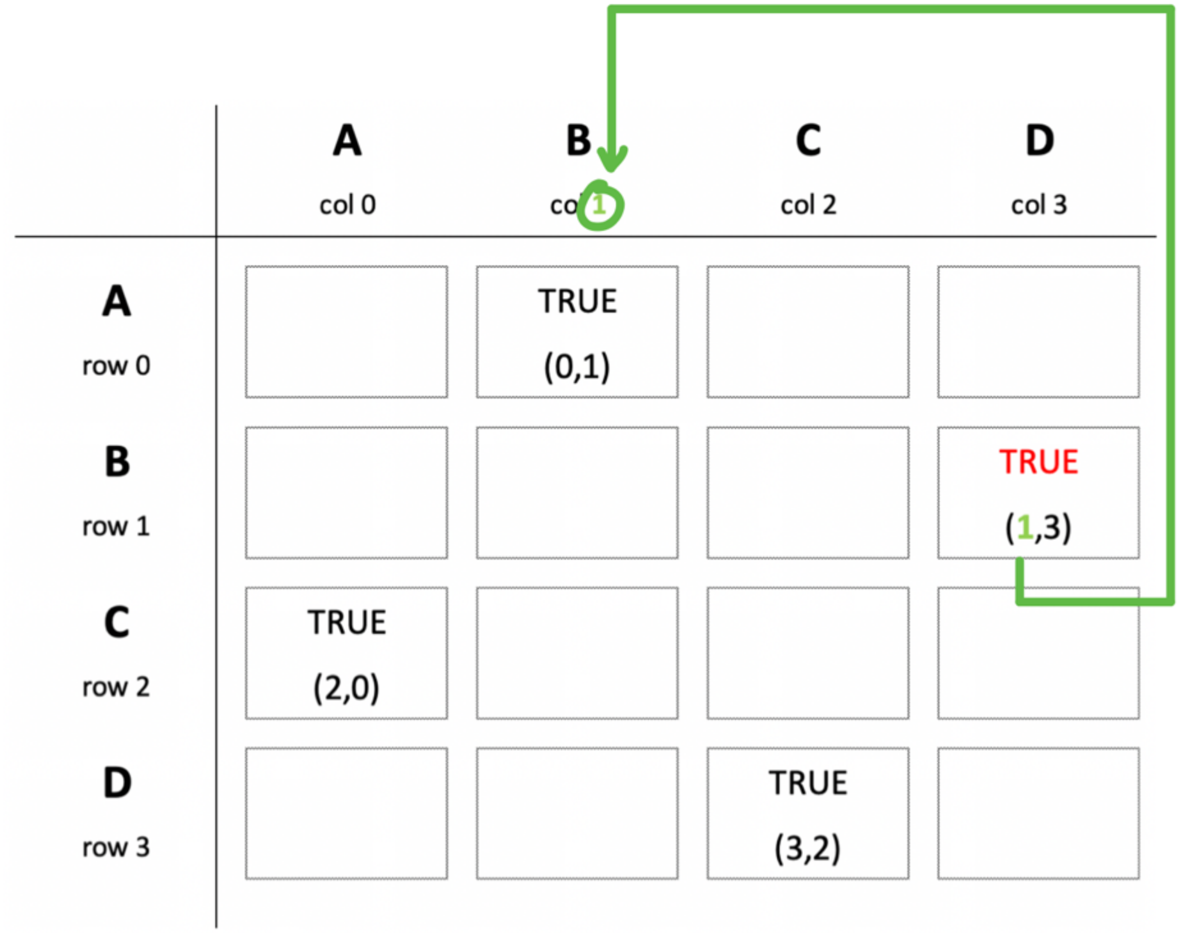


	A col 0	B col 1	C col 2	D col 3
A row 0		TRUE (0,1)		
B row 1				TRUE (1,3)
C row 2	TRUE (2,0)			
D row 3			TRUE (3,2)	

Alright! Let’s actually just add our B–>D pair for now and start checking!
If we happen to find a loop that would form by this, we can undo it and continue with the next pair.

Start of our check:

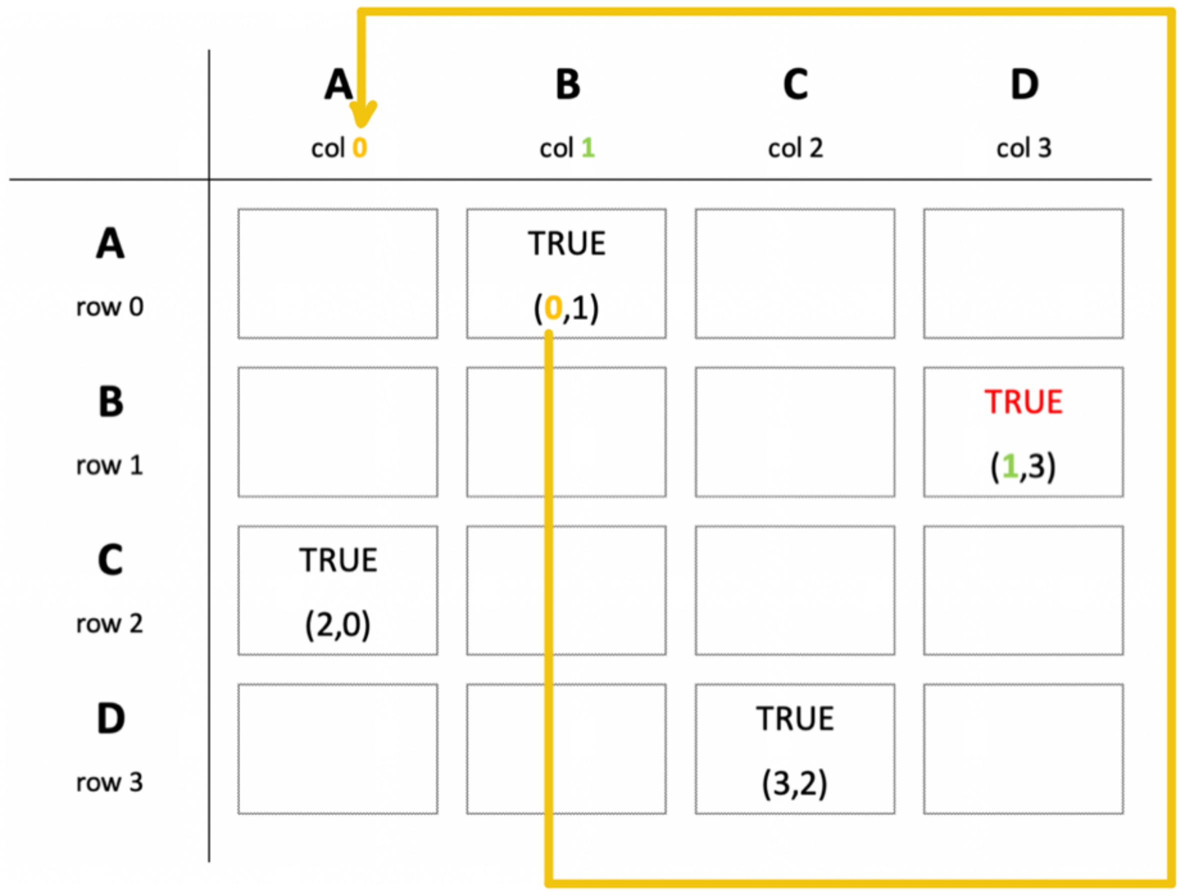
1.) So, we have been beat by B, as indicated by our first coordinate (1,3). Let’s look into that column i = 1 until we find another TRUE value, indicating that B has been beat.



	A col 0	**B** col 1	**C** col 2	**D** col 3
A row 0		TRUE (0,1)		
B row 1				TRUE (1,3)
C row 2	TRUE (2,0)			
D row 3			TRUE (3,2)	

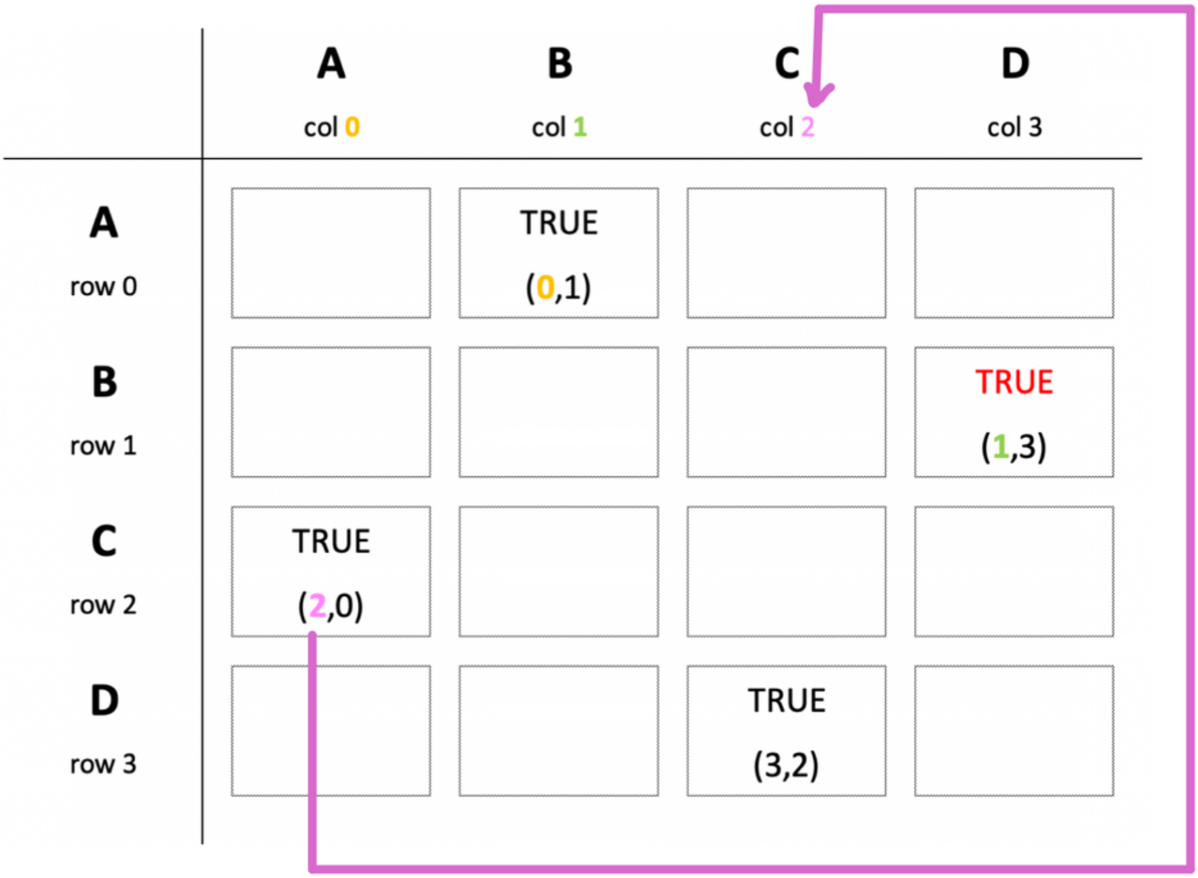
2.) Yep! In this case, we happen to find that B has been beat by candidate A. Let’s follow up column i = 0 for candidates who have beat A, to see if anyone of the already locked in edges indeed show, that they have beaten A.

Continue iterating through col 0 until we find any TRUE value.



	A col 0	B col 1	C col 2	D col 3
A row 0		TRUE (0,1)		
B row 1				TRUE (1,3)
C row 2	TRUE (2,0)			
D row 3			TRUE (3,2)	

3.) Here we go! We found a *TRUE* value indicating that A infact was beaten by anyone that has been found at row 2 for “candidates who beat C”. Setting this row’s value once again to our new ‘look in column i = 2’ value and look into that new column!

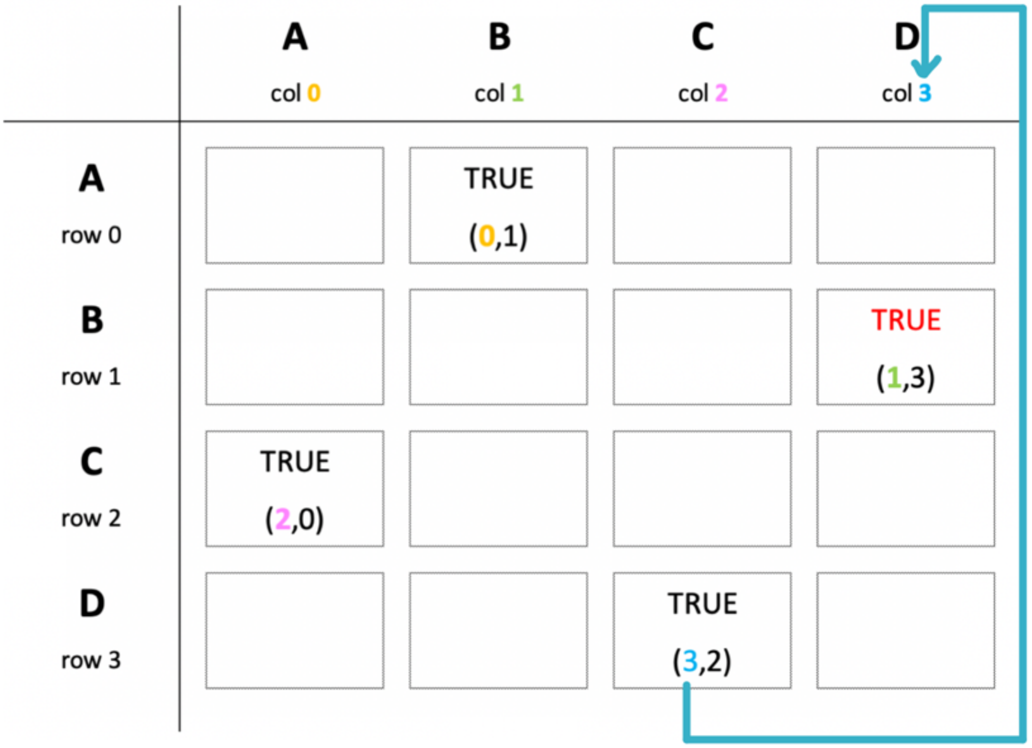


	A col 0	**B** col 1	**C** col 2	**D** col 3
A row 0		TRUE (0,1)		
B row 1				TRUE (1,3)
C row 2	TRUE (2,0)			
D row 3			TRUE (3,2)	

4.) At last in col 2 we found a *TRUE* value somewhere, that leads us back to to col i = 3 from where we started!

Because this is where we started we just stop because we know for sure that we will find a TRUE value in here, definitely creating a loop, if we have gone through all candidates!

That means: For *four candidates*, we actually started a column search *four times*. If, say we only needed 3 searches, there exists a candidate who has not been beat by those three. It is then save to add this edge.



	A col 0	B col 1	C col 2	D col 3
A row 0		TRUE (0,1)		
B row 1				TRUE (1,3)
C row 2	TRUE (2,0)			
D row 3			TRUE (3,2)	