## Found a false result from the constructor

SuffixTree { string: "xabxacxabbx", nodes: [ Node { start: None, end: Root, suffix link: None, children: [4, 6, 12, 8] },

- 1. Node { start: Some(2), end: Infinity, suffix\_link: None, children: [0, 0, 0, 0] },
- 2. Node { start: Some(2), end: Infinity, suffix\_link: None, children: [0, 0, 0, 0] },
- 3. Node { start: Some(3), end: Infinity, suffix\_link: None, children: [0, 0, 0, 0] },
- 4. Node { start: Some(0), end: Index(1), suffix\_link: Some(10), children: [1, 9, 0, 5] },
- 5. Node { start: Some(5), end: Infinity, suffix\_link: None, children: [0, 0, 0, 0] },
- 6. Node { start: Some(1), end: Index(1), suffix\_link: Some(0), children: [0, 10, 0, 7] },
- 7. Node { start: Some(5), end: Infinity, suffix link: None, children: [0, 0, 0, 0] },
- 8. Node { start: Some(5), end: Infinity, suffix link: None, children: [0, 0, 0, 0] },
- 9. Node { start: Some(7), end: Infinity, suffix link: None, children: [0, 0, 0, 0] },
- 10. Node { start: Some(2), end: Index(1), suffix link: None, children: [0, 11, 2, 0] },
- 11. Node { start: Some(7), end: Infinity, suffix link: None, children: [0, 0, 0, 0] },
- 12. Node { start: Some(2), end: Index(2), suffix link: Some(0), children: [0, 0, 3, 0] },
- 13. Node { start: Some(9), end: Infinity, suffix link: None, children: [0, 0, 0, 0] }], alphabet:

Node	Start	End	Children
0	None	R	[4,6,12,8]
1	2	8	None
2	2	8	None
3	3	8	None
4	0	i[1]	[1, 9, 5]
5	5	8	None
6	1	i[1]	[10, 7]
7	5	8	None
8	5	8	None
9	7	8	None
10	2	i[1]	[11, 2, ]
11	7	8	None
12	2	<i>i</i> [2]	[3, 13]
13	9	8	none

## I'll format it into a table:

The tree is inaccurate:

Go to node 4, map the node starting at index 0(x) and ending at index 1(a), Note that node 4 has 3 children.

Try to map node 4's child node 9.

From node 4, node 9 maps the characters of index 7(a) to infinity.

The issue is that now a path exists from root that represents 'xaabbx' which is not a substring of S.

Note though that really 4 should only have 2 children and that really node 9 should just represent just i[2] - i[2] with an internal node to represent bbx.

I'm not sure of a solution yet, I'm confused why this is the case.