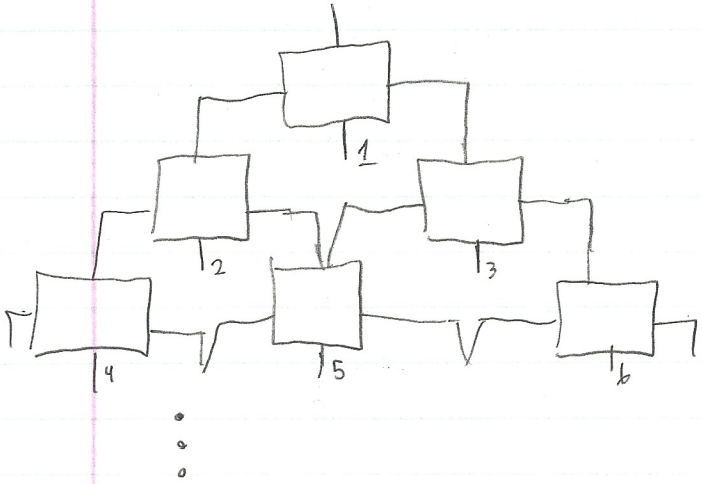
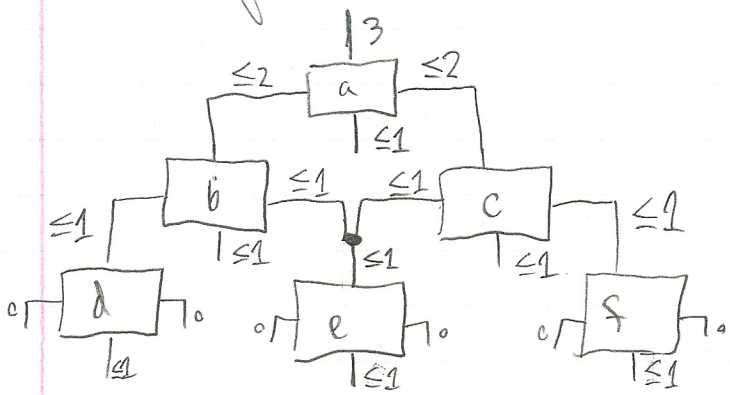
Shahid Nawaz sn1348

1 We will use $n(n+1)/2$ splitters arranged in a triangle as shown below. The bottom output will be the critical section or the unique name.



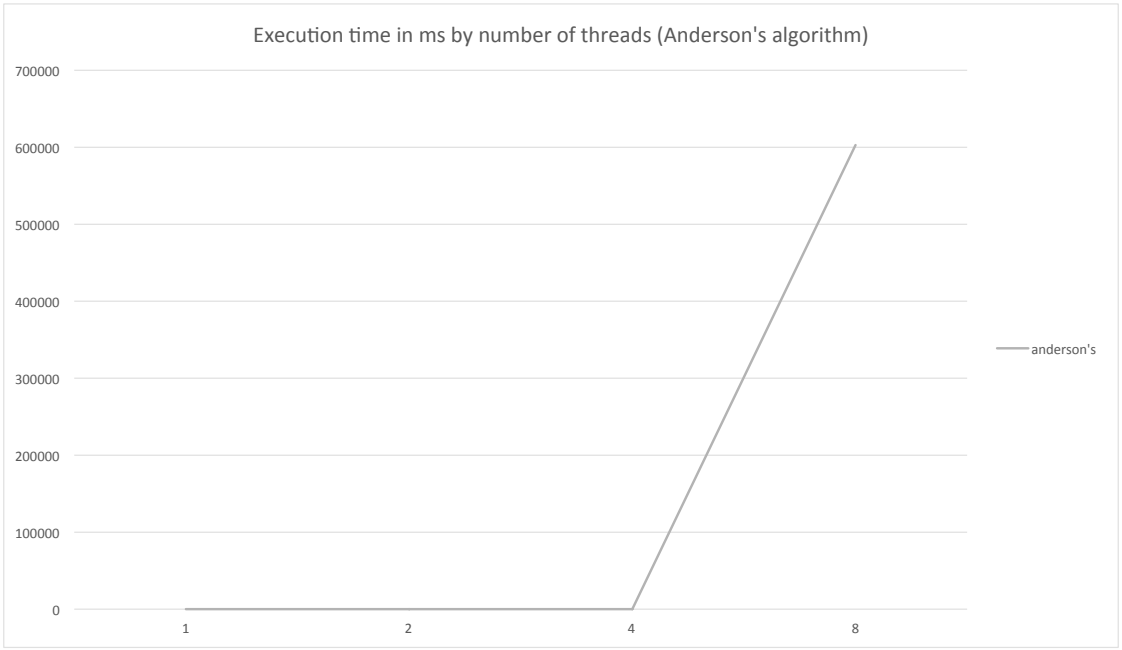
The question does not ask for a proof explicitly, so I will just show that it works for $n=3$.

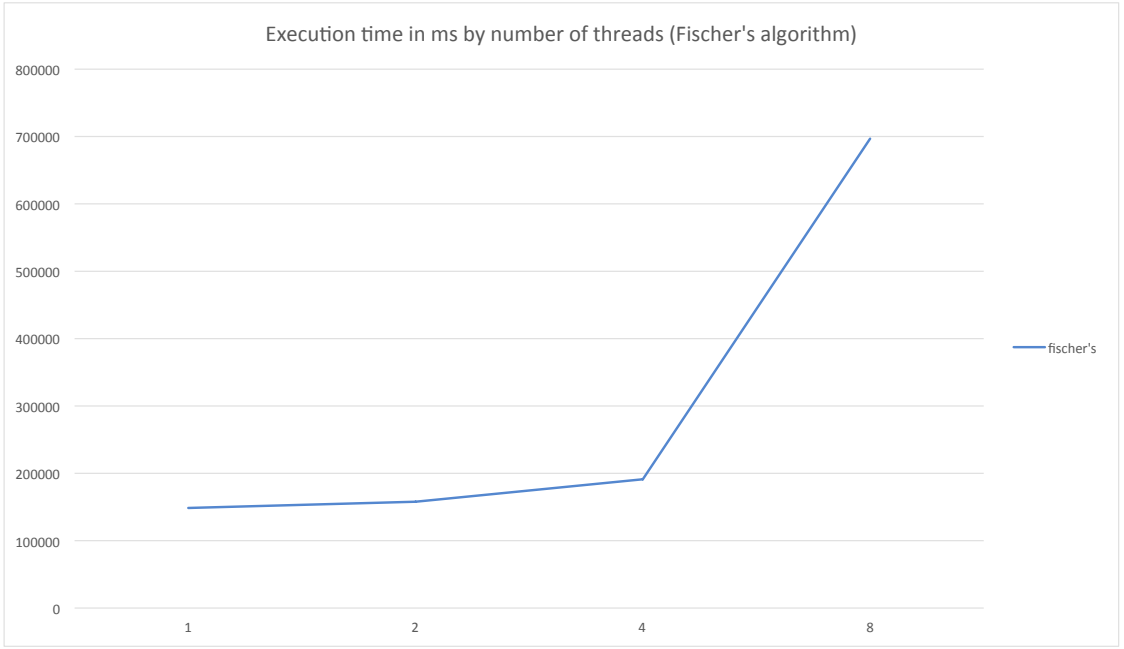


It is clear that this solution satisfies the index renaming property, any thread can end up anywhere, and it is independent of the last name. The inequalities describing the outcomes follow directly from the definition of a splitter, except for the last row. For the corners, it is obvious that at most 1 thread will get the C.S. and none will go to the sides, since at most one thread may enter. For the middle, a thread can come from either side, but they are mutually exclusive because for e to get one

From b, b must get 2 threads, from a but then only one thread is left for C. It will enter the CS. for C and not d to e. This holds in the other direction. Thus it satisfies the unique name requirement.

	numThreads	time (ms)
	1	148546
	2	157870
	4	191202
Fischer's	8	696829
	1	10
	2	17
	4	24
Lamport's	8	61
	1	33
	2	33
	4	60
Anderson's	8	602871




Execution time in ms by number of threads (Fischer's algorithm)
800000
700000
600000
500000
400000
300000
200000
100000
0
1
2
4
8
fischer's


Execution time in ms by number of threads
70
60
50
40
30
20
10
0
1
2
4
8
lamport's

# of threads	Time (s)
0	0.013656139
1	0.014465094
2	0.011657
3	0.012183905
4	0.01189208
5	0.012346029
6	0.012008905
7	0.012016058
8	0.011857986