PUMPING ARGUMENT

Starring Alice & Bob

ALICE & BOB

ALICE & BOB

My Baa-Baa language consists of all strings with more *a*s than *b*s.

ALICE & BOB

And I just invented
a finite state automaton
for your Baa-Baa language.

ALICE & BOB

ALICE & BOB

How big is it?

ALICE & BOB

It is really complicated!
It has a whopping 9999 states.

ALICE & BOB

Impressive!

So, let's test it on something big, but simple, like $b^{4999}a^{5000}$.

What does it say?

ALICE & BOB

No go.
It gets rejected.

ALICE & BOB

Well, then, your machine is faulty, since my test case has more *as*.

ALICE & BOB

Oops!

Sorry, there was a tiny bug.

Now it works fine.

ALICE & BOB

Okay, so tell me:

When does your machine first return to a prior state on my example, as you know it must?

ALICE & BOB

Oh.
I'll run it and watch this time.

After reading the 150*th* letter,
it's in the
same state it was
after the 50*th*.

ALICE & BOB

I see. So try your machine out on this bigger one: $b^{5099}a^{5000}$.

I'll bet you a fortune that it *will* accept it, even though it's ***not*** a Baa-Baa word.

ALICE & BOB

Not only that,
it will also fail on
$b^{5199}a^{5000}$,
$b^{5299}a^{5000}$,
ad infinitum.

ALICE & BOB

You're right!
How did you know, without even looking inside?

Give me a few minutes....

It's okay now.
I didn't even have to add any more states.

ALICE & BOB

I see.

So, now,
when does the
program
repeat itself?

ALICE & BOB

Oh.

After reading the second *a*
of $b^{4999}a^{5000}$,
it is in the
same state it was right after
the run of *b*s.

ALICE & BOB

Aha!

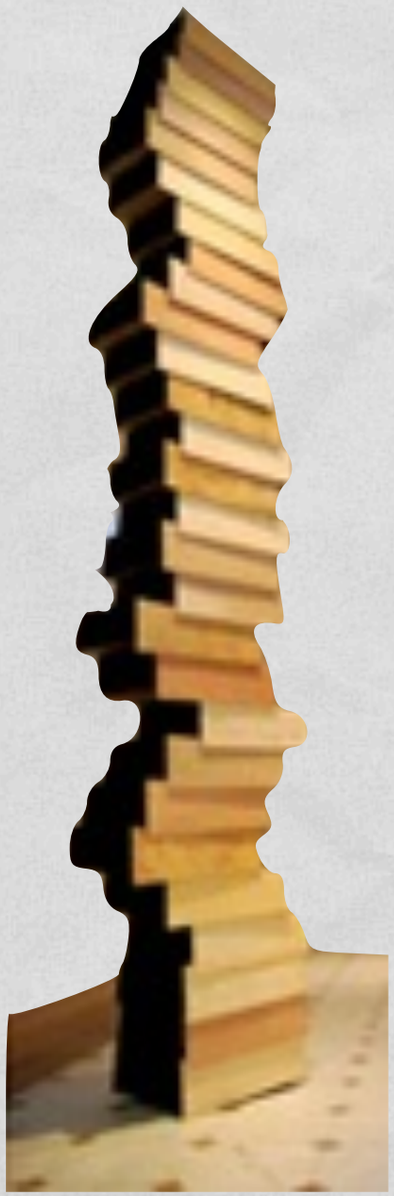
I can assure you, then, that it will give the ***wrong*** answer for $b^{4999}a^{4998}$.

ALICE & BOB

You're fantastic!

I better work on this some more.

ALICE & BOB



I think you will find that you need to add a stack to your program.

ALICE & BOB

Great idea.

You're a genius.

Thanks a million!

The End