Algoritmo per dire se un linguaggio regolare è infinito.

n è la costante del pumping lemma.

Si testano tutte le stringhe da lunghezza n a lunghezza 2n-1, se appartengono ad L.

Se troviamo anche una sola stringa, allora il linguaggio è infinito.

Perchè col pumping lemma possiamo pompare la stringa e ottenere sempre un'altra stringa appartenente al linguaggio

-------------------

Algoritmo per dire se linguaggio regolare contiene almeno 100 stinghe.

Prima si controlla se il linguaggio è infinito.

Se è finito, dal pumping lemma sappiamo che al massimo il linguaggio contiene stringhe di lunghezza n-1, altrimenti ci sarebbe un ciclo.

Si testano tutte le stringhe con lunghezza massima n-1, se appartengono al linguaggio si incrementa il contatore.

Alla fine si controlla il contatore.

---------------  
  
Algoritmo per dire se L = alfabeto, cioè contiene tutte le stringhe del suo alfabeto

Si considera il complemento di L.

Si considera il DFA, si invertono gli statia ccettanti e non accettanti.

Si usa un algoritmo di raggiungibilità di un grafo, se almeno uno stato accettante è raggiungibile dallo stato iniziale, allora FALSE, else TRUE

------------  
  
Algoritmo per dire se L1 e L2 hanno almeno una stringa in comune:

l'intersezione per i linguaggi regolari è chiusa.

Quindi L1 intersecato L2 = L3

Costruisco il DFA per L3.

Se trovo almeno un percorso dallo stato iniziale allo stato finale in L3, allora L1 e L2 hanno almeno 1 stringa in comune.

----------  
Algoritmo per dire se esistono delle stringhe nell'alfabeto tali che nn sono né in L1 ne L2

L3 = L1 unione L2 (sono chiusi)

Poi controllo il DFA di L3 complementato, inverto stati accettanti e non.

Se esiste almeno un percorso da stato iniziale a stati accettanti, allora TRUE, else FALSE.