(1) Given the grammar provided use the a^* algorithm to find the best parse for the example sentence. Show the agenda at each step. I've started you off...

	prob.	−log prob.
S → NP VP	1.0	0.00
VP → VM VV	0.9	0.04
VP → VV NP	0.1	1.00
VV → can	0.2	0.70
VV → fish	0.8	0.10
VM → can	1.0	0.00
NP → they	0.5	0.30
NP → fish	0.5	0.30

• 0 they • 1 can • 2 fish • 3

Agenda:

step 1: [NP, 0, 1] − 0.30

step 2: [VM, 1, 2] − 0.00
[NP, 0, 1] − 0.30
[VV, 1, 2] − 0.70

remember lowest cost = highest priority

step 3:
There's one more initialisation step before you're into the main loop...

when finished restart the video...

(2) how many steps did you save over doing the CKY algorithm?

a) Given the supertagger probabilities provided use the a* algorithm to find the best parse for the example sentence.

		prob.	-log prob.
can	S\NP	0.2	0.70
can	(S\NP)/(S\NP)	1.0	0.00
fish	S\NP	0.8	0.10
fish	NP	0.5	0.30
they	NP	0.5	0.30.

you can assume all non-stated probabilities are zero. — there's no point putting items of infinite cost in the agenda...

• they • can • fish •
0 1 2 3

always do the same thing for ties

Agenda:

step 1: [NP, 0, 1] - 0.3

step 2: [(S\NP)/(S\NP), 1, 2] - 0.00
[NP, 0, 1] - 0.30
[S\NP, 1, 2] - 0.70

step 3: ...

remember to calculate cost using $f(n) = g(n) + h(n)$