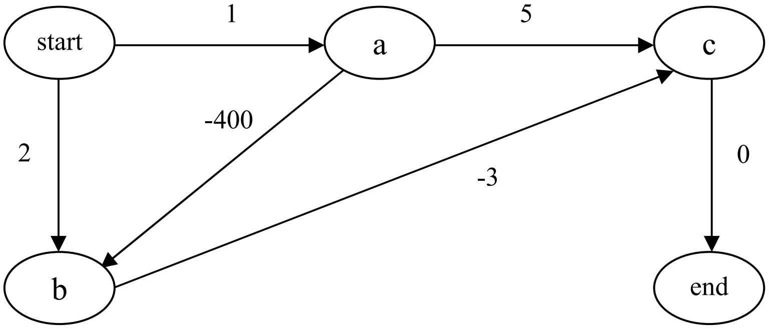
Problem 1 – Reinforcement Learning

Consider the *deterministic* reinforcement environment drawn below (let $\gamma = 0.1$). The numbers on the arcs indicate the immediate rewards. Once the agent reaches the 'end' state the current episode ends and the agent is magically transported to the 'start' state. The probability of an exploration step is 0.01.



a) Assume we choose not to use a function approximator (e.g., learning a neural network to output the Q-values) and instead learn a *Q-table…* a value for each arc in the above graph. Also assume all the initial values in your *Q* table are 8.

A learner follows the path *start* → *b* → *c* → *end*. Using two-step, standard *Q* learning, show the calculations that produce the new *Q* table entries and report the final *Q* table on the graph above.

b) If one could fit a *Q* table in memory, is there any reason to consider using a function approximator? Justify your answer. (If more than one reason, give the one you think is best.)

Name: ______________________

Problem 2 – Support Vector Machines

Consider a data set consisting of the following two examples:

F1	F2	Class
1	0	+1
1	1	-1

Assume the initial values of α_0, α_1, and b are all 0, and the C parameter is 10.

(a) Using a linear kernel <x,y>, show the first step of the SMO algorithm in which the alphas change.

(b) Show the same using a quadratic kernel, $(\langle x,y\rangle+1)^2$.

Problem 3 – Markov Networks

Explain why Markov networks cannot use the same parameter learning algorithm as Bayesian networks to compute the maximum likelihood parameters given the network structure and a complete data set (no missing data).

Name: ____________________

Problem 4 – Inductive Logic Programming

Suppose you wish to apply the ILP algorithm Aleph to a data set where each example is a directed graph.

(a) Using one positive and one negative example (draw two small directed graphs), illustrate your input to Aleph. Include examples and background knowledge.

(b) Assume the top of your refinement graph is the one atomic formula "positive(X)". Draw the next two levels of your refinement graph.

(c) What are Aleph's default scoring function and search strategy?