

Probabilistic Foundations of Artificial Intelligence
Final Exam

Date: 29 January 2013
Time limit: 120 minutes
Number of pages: 12

You can use the back of the pages if you run out of space. Collaboration on the exam is strictly forbidden.

[1 point] Please fill in your name and student ID:

1 [15 points] Search Strategies

- (a) **[5 points]** Consider a Social network (Facebook, for instance). You are given the names (unique NodeIDs) of A and B who are not friends with each other. Suppose you wish to find a shortest path through the network connecting A and B. To do so, you can ask any node in the network (whose NodeID you know) to reveal their list of friends (i.e., their NodeIDs). Which search algorithm would you use if your goal is to minimize the number of nodes asked? Explain the procedure and justify your choice.

- (b) **[5 points]** Consider the simplified road network of Switzerland depicted in Figure 1. The values on the edges denote the road distance between the two cities it connects. The task is to find the shortest path from Berne to Montreux through this road network. Table 1 gives the straight line distances between the various cities to Montreux. Using this table as a heuristic function, enumerate (by expanding the search tree, and annotate the order in which nodes are expanded) the steps involved in performing A* search on the given network to achieve the task of travelling from Berne to Montreux.

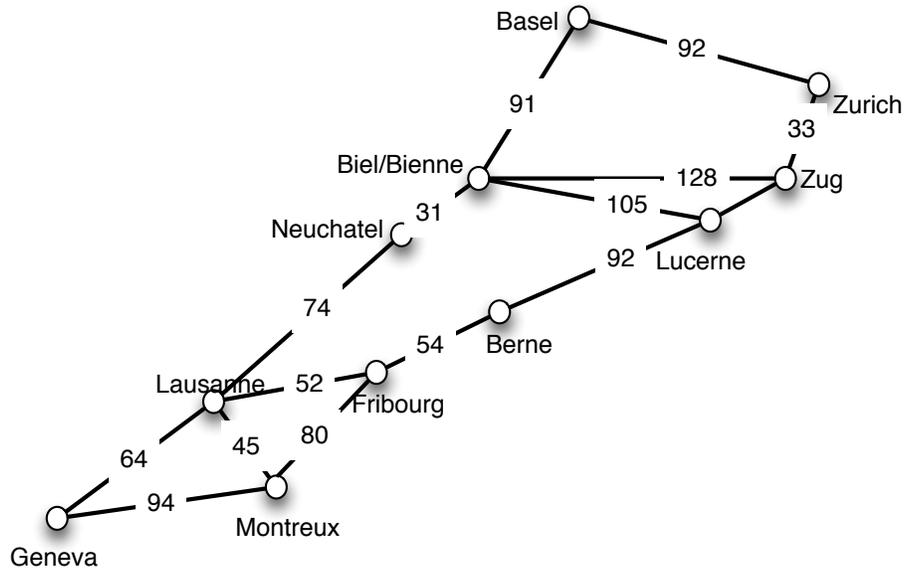
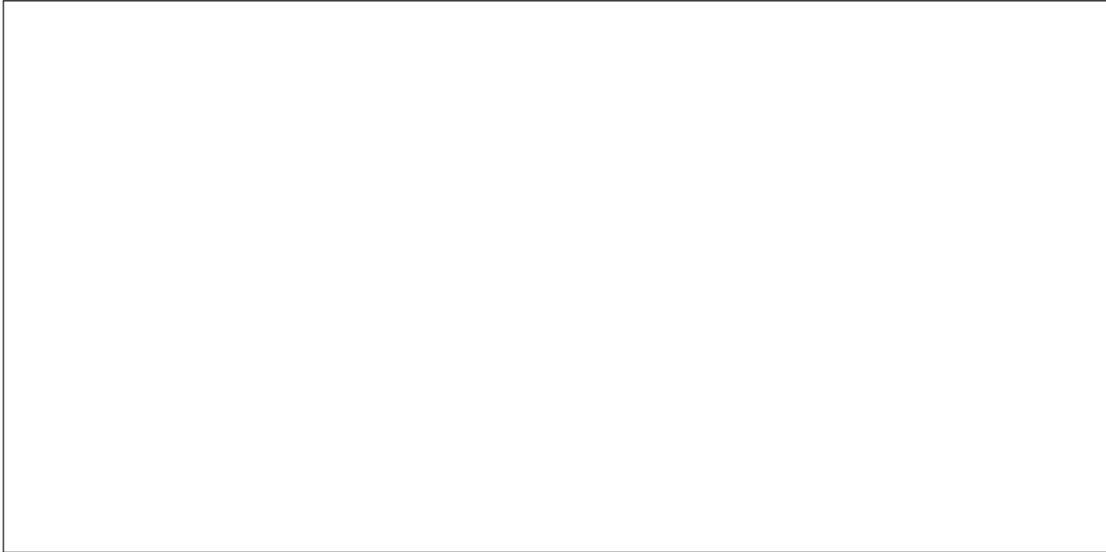


Figure 1: Simplified Road Network of Switzerland

Table 1: Straight Line Distances to Montreux

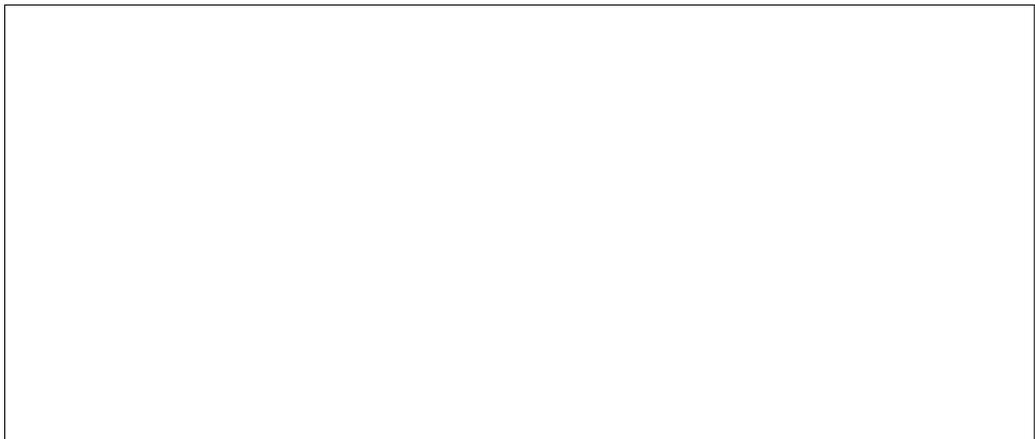
Geneva	65	Biel/Bienne	83
Lausanne	20	Zug	147
Friborug	44	Zurich	162
Berne	70	Lucerne	127
Neuchatel	62	Basel	135

- (c) **[5 points]** Now, instead of using distances (as in Figure 1), suppose we wish to use time of travel as the values on the edges between cities. Note that due to speed limits on various legs, the speed of the vehicle may be different for each segment. The speeds range from a minimum value V_{min} to a maximum of V_{max} . Assume that the time of travel is deterministic and is not affected by traffic or road conditions. Using the straight line distances in Table 1, suggest an appropriate non-trivial heuristic function for the new problem setup and justify your choice.

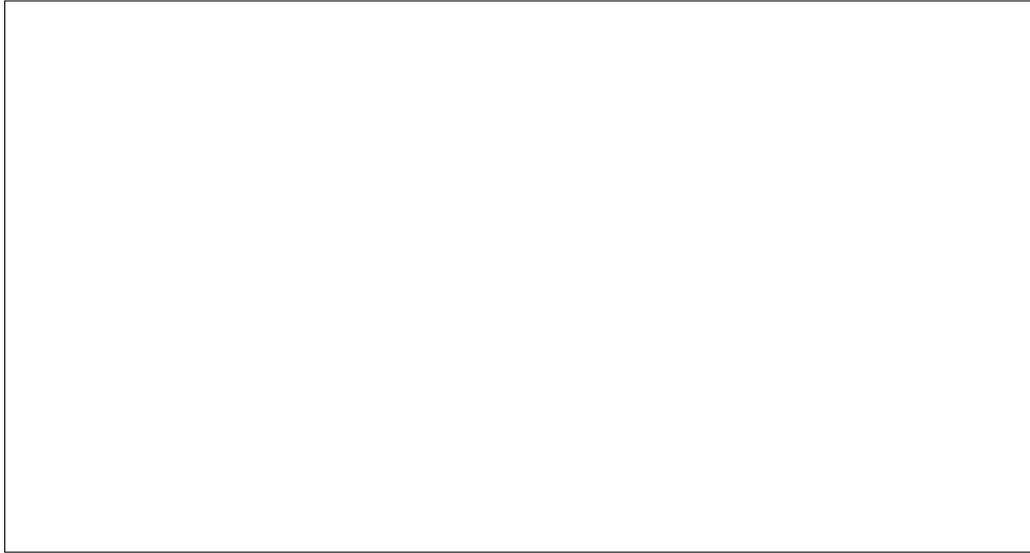


2 [15 points] Propositional and First-order Logic

- (a) Suppose there is a building with rooms in multiple floors. To model the world in first-order logic, we use $Above(x, y)$ as the predicate indicating that room x is on the floor which is higher than the floor of room y , and use $Level(x, y)$ as the predicate indicating that x and y are on the same floor of the building. Using connectives, quantifiers and precisely the predicates $Above$ and/or $Level$, answer the following two questions.
- (i) **[5 points]** Translate the sentence “(Room) A is on the top floor of the building.” into first-order logic.



(ii) [5 points] Using first-order logic, state that the building has exactly two floors.



(b) [5 points] Suppose A , B and C are propositional symbols. Prove using resolution that if $A \Rightarrow B$, $B \Rightarrow C$, and $C \Rightarrow A$, then it holds that $C \Rightarrow B$.



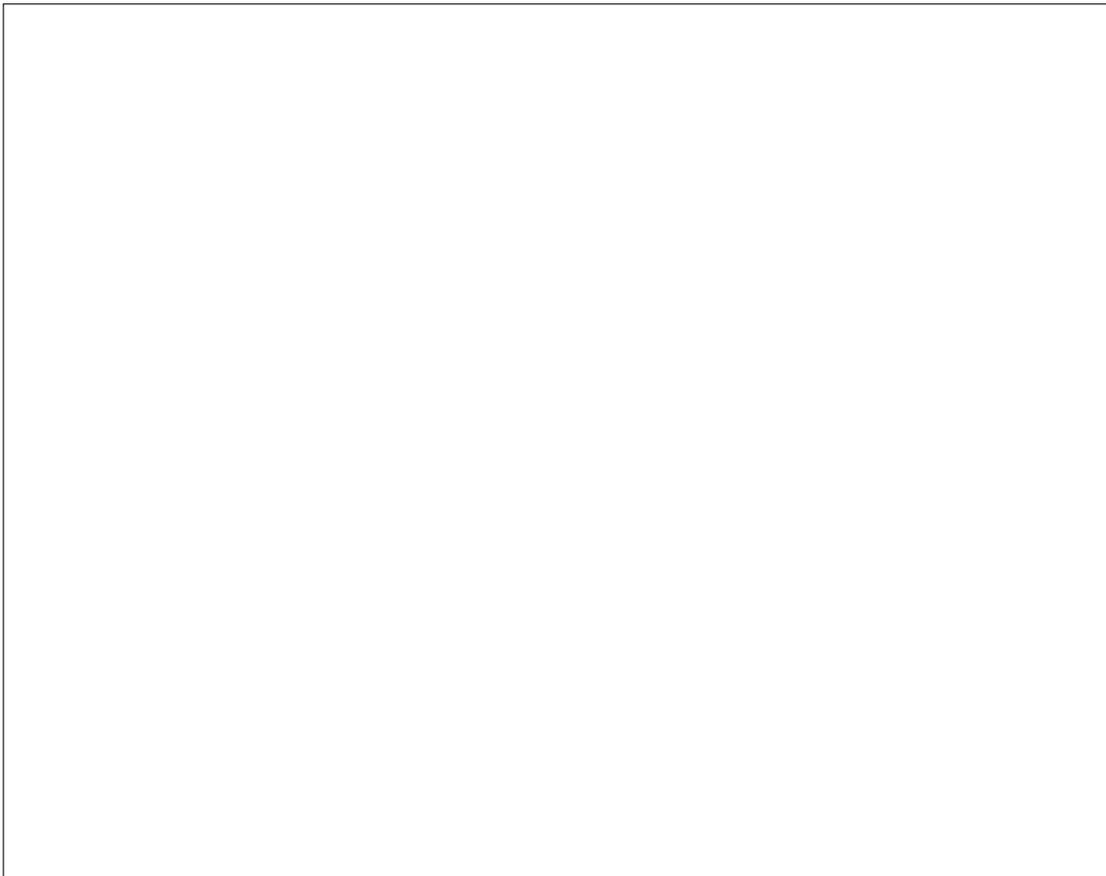
3 [15 points] Dinner Dilemma

Tired of breaking your diet time and again, you decide to leave it to chance to decide your dinner. There are two boxes, A and B. A contains 3 black balls and 6 red balls, B contains 4 black and 4 red balls. Every night you pick a box randomly (0.5 probability each) and pick one ball from the box uniformly at random, replacing the ball into the same box such that the setup remains the same for future nights. Your dinner is decided as a combination of ball colour and the current weather. Assume that $P(\text{Weather}=\text{clear}) = 0.4$ and $P(\text{Weather}=\text{overcast}) = 0.6$. Table 2 describes this selection procedure.

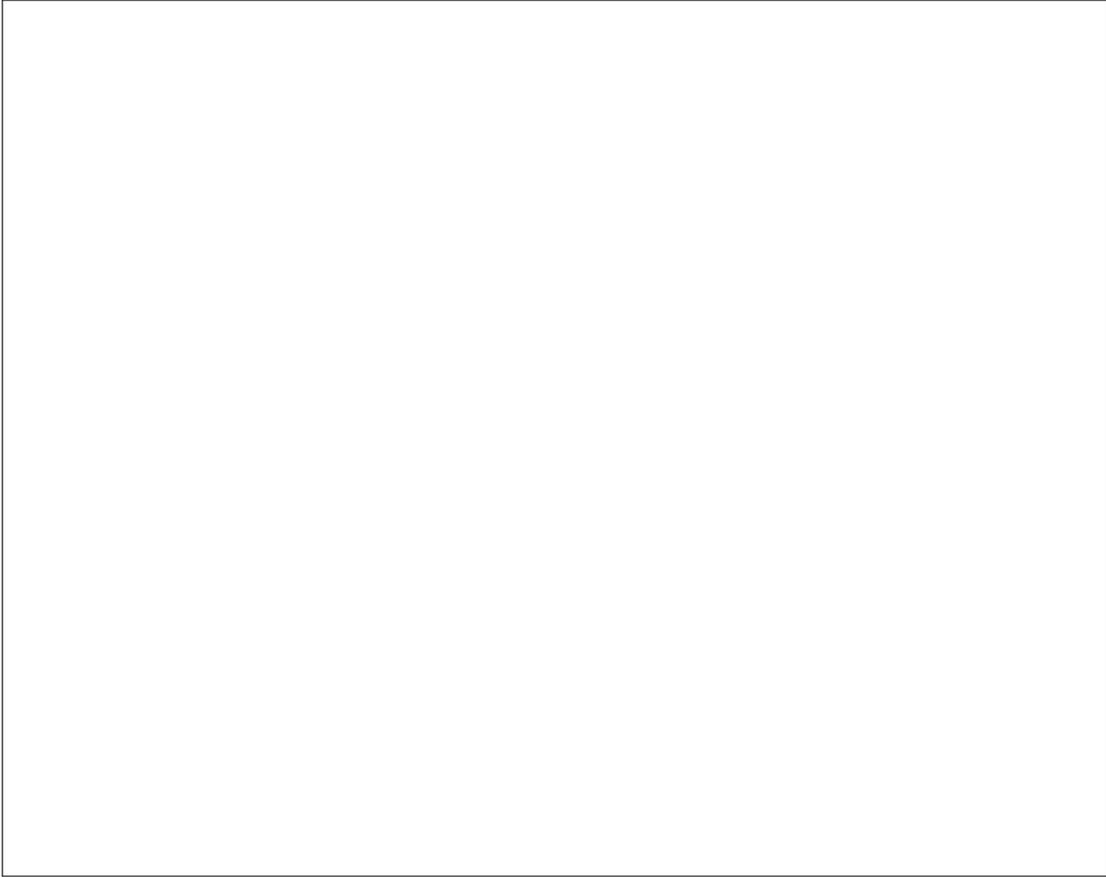
Table 2: Dinner Decision

Ball Colour	Weather	Dinner
red	clear	pizza
red	overcast	salad
black	clear	salad
black	overcast	pizza

(a) [3 points] Draw the Bayesian Network corresponding to this decision problem



- (b) [7 points] Using the Bayesian Network you have constructed and the probability values given in the problem, compute the probability that **box A** was picked given that the weather was **clear** and the colour of the ball picked was **red**.



- (c) [5 points] Given that a **red** ball is picked, compute the probability that the weather is **overcast**



4 [10 points] Bayesian Networks: d-separation

Consider the Bayes net below. Using the notion of *d-separation*, explain whether the following statements of conditional independence hold or not.

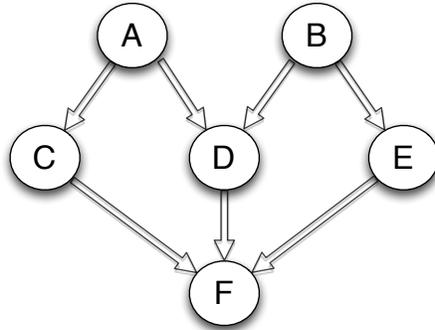


Figure 2: Bayes net

(a) [5 points] $A \perp B \mid C, E$

(b) [5 points] $A \perp F \mid C, D$

5 [20 points] Vacuum Cleaning Robot

Consider the following problem of probabilistic planning. A vacuum cleaning robot cleans a house based on its battery power. Assume that the robot can only perceive its battery level (its state) as *high*, *low* or *drained*. As long as the battery has not run out, the robot can choose (its actions) to either *clean* the house, or *wait*. When the battery level is *drained*, the robot can only *wait*. Assume the discount rate γ of this MDP is $1/2$.

- (i) If the robot waits, the battery level will stay the same and the robot receives no reward.
 - (ii) If the robot cleans and the battery level is high, it will drop to low with probability $1/2$; if the robot cleans and battery level is low, battery will drain out with probability $1/2$.
 - (iii) If the battery doesn't drain out when cleaning, the robot receives a reward of 2; otherwise it receives a reward of -4 .
- (a) [7 points] Draw the MDP described above, annotating the action-dependent transitions with transition probabilities and associated rewards.



- (b) **[3 points]** Let π denote any policy. Recall that the value function $V_\pi(x)$, denoting the long term expected future reward when starting in state x and following policy π , is given by

$$V_\pi(x) = \sum_{x'} P(x'|x, \pi(x)) [r(x, \pi(x), x') + \gamma V_\pi(x')].$$

Let V^* denote value function associated with the *optimal* policy π^* . Briefly explain why $V^*(drained) = 0$.

- (c) **[10 points]** In order to compute the optimal policy for the given MDP, we consider the policy iteration approach here.

Recall that policy iteration starts with an arbitrary (e.g., random) initial policy π . Until convergence, it iteratively computes the value function $V_\pi(x)$ for the current policy and then updates the current policy to be the greedy policy π_g w.r.t. the computed $V_\pi(x)$. The greedy policy for a value function is given by

$$\pi_g = \arg \max_a r(x, a) + \gamma \sum_{x'} P(x'|x, \pi(x)) V_\pi(x')$$

Compute the optimal policy and its value function for the above MDP. [*Hint:* To save time, start with an initial guess for the optimal policy and prove that it's greedy *w.r.t.* the corresponding value function, i.e., policy iteration terminates.]

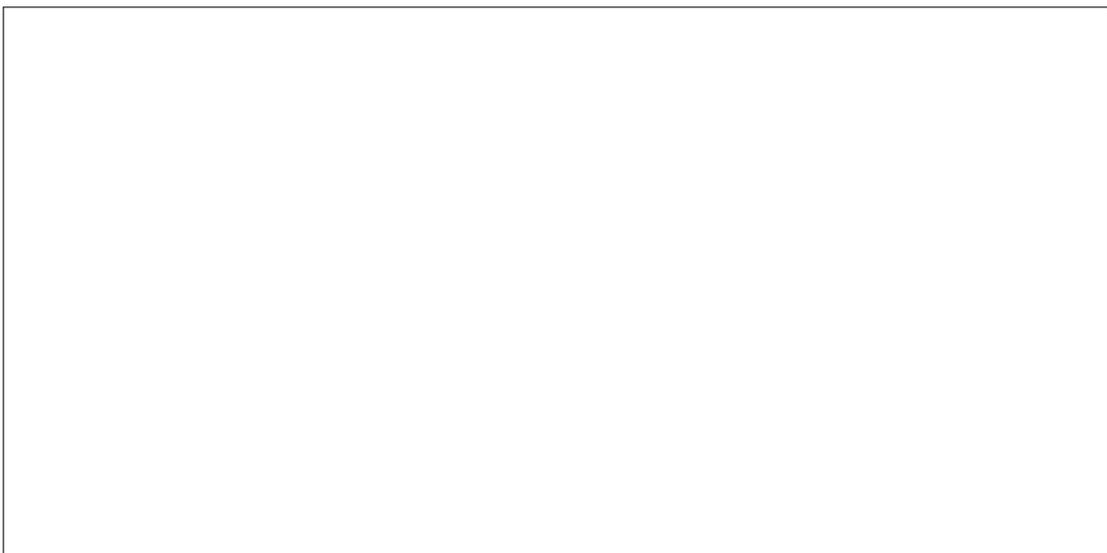
 Continue to
next page for
more space



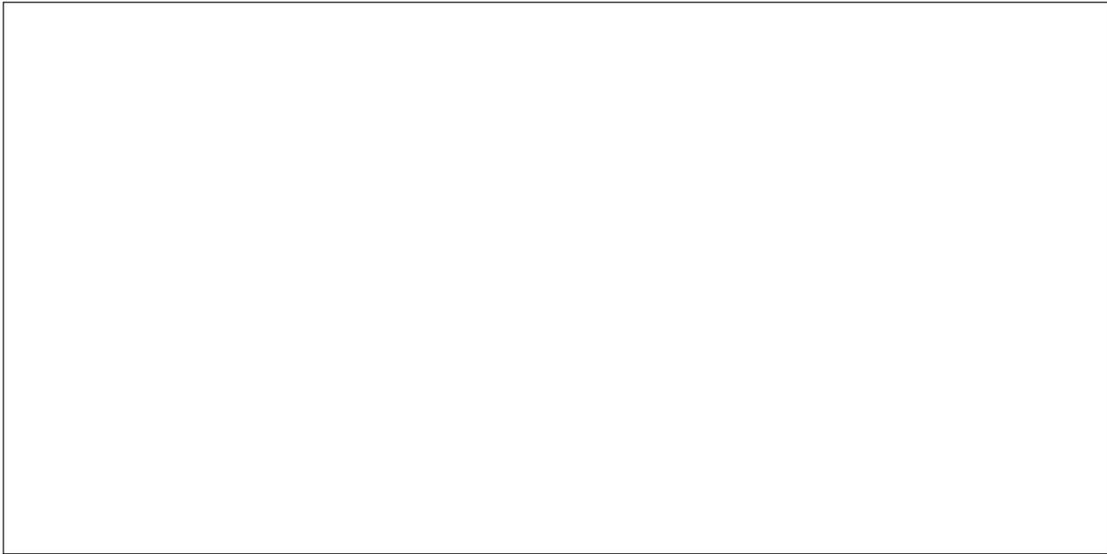
6 [12 points] Information Gain

Consider three discrete random variables A, B, C . State whether the following statements are true, false, or cannot be inferred from the information given. Explain your answer. (I refers to the information gain and H refers to the entropy function.)

(a) [4 points] $I(A; B) \leq I(A; B, C)$



(b) [4 points] $I(A; B) + I(A; C) \geq I(A; B, C)$



(c) [4 points] $H(A) - H(A|B) + H(C) \geq H(B) - H(B|A) - H(C)$



7 [12 points] K -armed bandits

Suppose you go to a casino and decide to gamble on a row of K slot machines. Each slot machine a_i is associated with a probability μ_i of getting a reward 1 and probability $1 - \mu_i$ of getting a reward 0. Your goal is to maximize the total reward. However, the reward probabilities $[\mu_1, \dots, \mu_K]$ are unknown to you and you face the problem of which machines to play and in what order.

The goal is to learn the policy of playing the actions (pulling a particular slot machine) so as to maximize the expected reward you get. We assume that the state of the slot machines do

not change and hence the rewards observed from playing action a_i are *i.i.d.* with expected value of μ_i . Use γ to denote the discount factor.

- (a) **[4 points]** Draw the MDP for a $K = 3$ armed bandit problem. Use the variables a_i for annotating actions and μ_i for unknown rewards.



- (b) **[8 points]** Write down the pseudo-code for *Optimistic Q-Learning* algorithm for playing K -armed bandits. Use α_t to denote the learning rate in your algorithm.

Recall that *Optimistic Q-Learning* initializes the reward estimates for each state-action pair to a high value (denote by R in your pseudo-code). At each iteration, it picks the action a with the highest estimated Q-value from all the available actions in the current state x .

