ECI- optimization under uncertianty

Exercises –A.Marchetti-Spaccamela

Monkey for food

We have discussed in class a 9 competive deterministic algorithm. The algorithm we discussed was in rounds; at round i the monk was traveling 2^i in one direction; if she did not find food then monkey returns to the initial point; and then during round (i+1) she walks 2^(i+1) in the opposite direction. Note that the monk doubles each time the distance from the orgin.

a) Analyse the expected competitive ratio of the algorithm in which the monk randomly chooses the direction to start.
b) Assume that instead of two possibilities (right/left) there are three direction to be explored. Design and analyse an algorithm for this case.

Space filling algorithm for TSP

With reference to the space filling algorithm for TSP ansie:

a) If carrying capacity is not a limitation, a single truck can make all the required deliveries in fewer miles than can a fleet of trucks, but will take longer. Explain why.
b) Assume that there are three vehicles that must traverse the tour (we have seen in class the case of one tou). Extend the algorithm to this case trying to to balance the length among the three drivers?Is your algorithm still a log N approximate? Explain your answer.
c) Can the spacefilling curve heuristic be used to route a vehicle in a city that has a large lake or park through which traffic cannot pass? What problems might this cause the routing system? Explain.

Mistake-bound model

k-CNF is the class of Conjunctive Normal Form formulas in which each clause has size at most k. E.g.,

$$x_4 \text{ and } (x_1 \text{ or } x_2) \text{ and } (x_2 \text{ or } x_3 \text{ or } x_5)$$

is a 3-CNF.
Give an algorithm that learns the formula in the mistake bound model. Namely, the algorithm is given a sequence of truth assignment of a unknown 3-CNF formula over n boolean variables and must guess whether the assignment satisfies the formula or not. Then the algorithm is said whether the answer was correct or not.
Propose an algorithm that each time guesses the vaue of the formula and makes as few errors as possible. Your algorithm should run in polynomial-time per example (so the “halving algorithm” is not allowed). How many mistakes does it make at most?