When I went to school (University of Manitoba), they (Instructor of the Online Algorithms course) asked me what I wanted to be when I grew up (graduate from the university). I wrote down ‘happy’. They told me I didn’t understand the assignment (the one below); and I told them (anonymously, on Piazza) they didn’t understand life . . . John Lennon (almost)

Please pay attention to the followings when preparing/submitting your assignment:

• All problems are written problems. There are seven problems with a total of 113 marks plus 25 bonus marks. To get a complete mark, you need to collect 110 marks.

• Do not approach the bonus problems before finishing work on the mandatory parts. Bonus problems are harder and you are not expected to know them for your final exam.

• If you have any question related to the assignment, you are encouraged to post it on Piazza. Note that you can submit anonymously (in case you are shy). The instructor is likely to drop hints related to your questions when it is posted on Piazza (because all students can take advantage). It is not the case when you ask questions in emails or during office hours. Do not post any answer or hint on Piazza (for the obvious reason).

• You are welcome to discuss the problems with your friends (or enemies). But you should write your answers individually.

• Submit your answers electronically using UMLearn. You should submit a single pdf file. You are encouraged to prepare your assignment using \LaTeX.

• If you feel the assignment is too hard (or too simple), do not panic! To some extent, this assignment serves to indicate how easy/hard the future assignments/exam should be.

Problem 1 Ski-rental & Randomization [10 marks]

Consider the following algorithm for the ski-rental problem: flip a coin at the beginning, if it is a tail, buy the equipment at the beginning; if it is a head always rent and never buy. Assume the cost of buying is $b$ and the cost of renting is 1 per day; let $x$ denote the number of days that the player goes skiing.

a) What is the expected cost of the algorithm in terms of $b$ and $x$?

b) Is the algorithm competitive? You need to either prove a constant bound for the competitive ratio OR show via adversarial input that the competitive ratio is not constant.
Problem 2 Path-cow Problem [15 marks]

Assume the path-cow problem where the cow faces a \( w \)-way path, i.e., a star of \( w \) rays. Note that when \( w = 2 \), the problem becomes our classic path-cow problem. The cow is very inspired by the doubling technique that we learned in the class and uses it. This means that she starts with the first ray, goes 1 unit in that direction, comes back to the origin, then 2 units in the direction of ray 2, back to origin, 4 units to the direction of ray 3, back to origin, and this continues until it finds the hole. We learned in the class that when facing \( w \) rays, the cow should make jumps of \( \frac{w}{w-1} \), which is less than 2 for \( w > 2 \). So, our cow is not always doing well here. To make sure, we verify it here:

a) Assume the cow finds the hole in her \( k \)'th move and at distance \( u \) from the origin. What is the total distance moved by the cow?

b) Where does the adversary place the hole in order to harm the algorithm?

c) What is the competitive ratio of this algorithm? (Hint: you should express the ratio in terms of \( w \)).

Problem 3 Online Bidding [10 + 15 marks]

a) Consider the bidding algorithm in which bids are powers of a real value \( \gamma \), i.e., \( 1, \gamma, \gamma^2, \ldots \). The case with \( \gamma = 2 \) becomes the doubling algorithm. Follow the same steps take in class to provide a formula for competitive ratio in terms of \( \gamma \).

b) (bonus) Assume we select a number \( \gamma \), randomly from range \((0, b)\) where \( b \) is a given real value. Then the bid of the algorithm are \( 1, \gamma, \gamma^2 \ldots \). Follow the steps that we took for analysis of the randomized bidding algorithm to prove an upper bound for competitive ratio of this algorithm.

Problem 4 Clustering & Advice [25 + 10 marks]

Considering the online clustering problem. Recall that in this problem, we need to partition a sequence of online points into \( k \) clusters so that the maximum diameter of clusters is minimized.

a) Show that \( O(n \log k) \) bits of advice are sufficient to achieve an optimal clustering.

b) The competitive ratio of the best deterministic algorithm that we saw in the class was 8. Consider inputs for which the cost of \( \text{OPT} \) is a large value. Show that for any such input \( \sigma \), provided with \( O(\log(\text{OPT}(\sigma))) \) bits of advice, we can achieve a competitive ratio of 2. Here, \( \text{OPT}(\sigma) \) is the cost of \( \text{OPT} \) for \( \sigma \). To get the full mark, you need to precisely indicate i) what the advice encodes? ii) how the algorithm works? iii) why the algorithm is 2-competitive?

c) (bonus) Assume we are just given a constant number of bits. What advice should be to improve the competitive ratio to something smaller than 8?

Problem 5 List-Update Algorithms [10 marks]

Consider the Move-To-Front-Every-Other-Access algorithm for the list update problem. Here, an item \( x \) is moved to front on every other access to it, i.e., on the second, forth, sixth, etc. accesses to \( x \). Use an adversarial argument to show that the competitive ratio of the algorithm is at least 2.
Problem 6  List Update Algorithms [25 marks]

a) Consider a variant of the list update problem where making a paid exchange has a cost of 2 instead of 1. Show that the competitive ratio of Move-To-Front remains at most 2.

b) Consider another variant of the list update where accessing an item at position $i$ has a cost of $2i$ (paid exchanges have a cost of 1 as before). Prove that the competitive ratio of Move-To-Front in this variant model is at most 4. (Hint: you need to change the definition of the potential function).

Problem 7  Compression [18 marks]

a) Apply the Burrows-Wheeler transform on the following string; show your work and the output

$$ALIBABA$$

Assume $\$ \$ precedes all characters when you sort rotations.

b) Assume an initial list $\$ \rightarrow A \rightarrow B \rightarrow C \rightarrow D$, i.e., initially $\$ is at index 0, $A$ is at index 1, etc. Assume we use Move-To-Front on the above list to encode $BADBABA\$$. Show what numbers that will be encoded.

c) Assume an initial list $\$ \rightarrow A \rightarrow B \rightarrow C \rightarrow D$. A compressing scheme that uses Move-To-Front has encoded the following numbers for a text $T$. Show what the actual text is. The numbers are $3 \ 0 \ 2 \ 1 \ 0$. 
