Advanced Algorithms Ex.3

Deadline: July 9

Note: Questions that start with ☺ are not for submission and are given as warm-up and for the sake of your confidence with the materials, but be aware that you are supposed to know how to solve them.

1 Online algorithms analysis

Memory Game

In *Memory Game* for one person, there is n pairs of matching cards. The cards are all with face down, but contain pictures of objects when face up, where each object appears in exactly 2 cards (a pair). In each round, you can turn two cards face up to reveal their pictures. If the pictures match - the pair removed from the game. If they don't match - they turned down, hiding their pictures once again. The game ends when all pairs are removed, and the score is the amount of rounds needed to do so. Suppose that you can remember the picture on every card that you have seen.

- Show a deterministic algorithm for *Memory* with competitive ratio of 2.
- Show a lower bound of $(2 - \frac{2}{n})$ for competitive ratio of any deterministic algorithm for *Memory*.
- Give a random algorithm which per round chooses the first card uniformly at random from the unseen cards, and show it has an expected competitive ratio of $1\frac{3}{4}$ against an oblivious adversary.

Random Path Search

In class we have seen an algorithm which is 9-competitive for the Path Search problem.

Describe a random algorithm which his expected competitive ratio against an oblivious adversary better than the ratio you have seen in class.

Random Rental ski

- Show a random algorithm for rental ski, with better competitive ratio against an oblivious adversary than discussed in class (which was about ≈ 1.809).

2 Stringology

Periodicity

A *period* of a string is a prefix that can be used to generate the whole string by repeating the prefix (the last repetition may be partial). For a string S of length n, the *periodicity* of S is the shortest period of S and also defined as smallest index p such that $S[i] = S[i+p]$ for all $i \leq n - p$.

Example: for $S = abcabcabcab$, abc and $abcabc$ (and more) are periods of S, and S's periodicity is 3 (Note $S = (abc)^{3\frac{2}{3}}$).

- Give an algorithm that make use of the automaton of KMP algorithm and find the periodicity of a given string S, in $O(|S|)$ time.
- ☺ Give an algorithm that gets as input an LCP data-structure of a string S, and find the periodicity of S in $O(p)$ time.

Palindromes

A palindrome is a string A such that $A = A^r$, when A^r is the string A written in reversed form. Let S be a string of length n over an alphabet bounded by n.

- ☺ Give a linear (both time & space) algorithm that finds a longest substring of S A, where A is a palindrome.
- Give a linear algorithm that finds a longest substring of S A, where A is a palindrome, among the palindromes which occur in S exactly once.

You can use the following theorem.

Theorem 1 (Farach 1997, link)) *Given a string $S \in \{1, ..., n\}^n$,the suffix tree of S can be deterministically constructed in $O(n)$ time and space.*