Assignment 2.

Post date: Sept. 23, 2014.

Due date: Sept. 29, 2014 (Monday) by noon time. **The answers need to be typed up with a computer.** Hand in printed paper at assignment box at 4th floor of MC across from the tutorial centre.

Question 1. (10 marks)

In order to implement the LRU (Least Recently Used) algorithm for paging, a data structure is needed to maintain the page ID and the least recently used time stamp for each page in the cache. The data structure should also quickly return the page that has the oldest time stamp. Describe a data structure that maintains such a cache index and supports the following two operations:

1. isExist(p): Check if a page ID p exist in the cache.
2. stamp(p, t): Change the stamp of page p to the current time t. It is known that the new time t is always larger than the old time stamp of p.
3. oldest(): Returns the page with the oldest time stamp.

The data structure should have an average $O(1)$ time complexity for each of these operations.

Question 2. (20 marks)

In the Secretary problem (selecting the largest diamond) in the course note, the $k = \frac{n}{e}$ strategy is an asymptotically optimal choice for large n. But for small n, the best value of k can be computed too.

(A) Let $P(n, k)$ be the success probability of the strategy of observing the first k diamonds out of n diamonds. Design an algorithm that computes $P(n, k)$ for all $k = 1,2, \dots, n$ in $O(n)$ time. You can use every property we proved in the course note by citing the fact without proof.

(B) Note that the answer to (A) is naturally an algorithm to compute the optimal k given n. But if you repeat this for every $n = 1, \dots, N$, it takes $O(N^2)$ time to compute the optimal k for every n. Now design an $O(N \log N)$ time algorithm to compute the optimal k for every $n = 1, 2, \dots, N$. You can assume the following lemma is true without proof.

Lemma: Let $f(n, k) = k \cdot \sum_{i=k}^{n-1} \frac{1}{i}$. Given n, suppose $k = k_{opt}$ is such that $f(n, k)$ is maximized. Then

1. $f(n, k-1) < f(n, k)$ for every $k < k_{opt}$, and
2. $f(n, k) > f(n, k+1)$ for every $k > k_{opt}$.