COL758: Advanced Algorithms **Spring 2019**

Lecture 23: April 18

Lecturer: Naveen Garg *Scribe: Rahul V*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Brief: In this lecture, we discuss the competitive ratio of the MARKING algorithm and the lower bound for randomized algorithm. Random walk on complete graph is studied in short, and the results are borrowed, alongside the Yao's Theorem to prove that lower bound for randomized algorithm is $\log k$.

23.1 Competitive ratio of MARKING Algorithm

Consider k distinct page requests where $k = 100$ is the size of the table. Let c of them be request to clean pages, i.e., pages that are not in the cache at the start of the phase and the remaining $k - c$ are requests to pages in cache at the start of the phase. Let $c_1 c_2 c_3 s_4 s_5 c_6 s_7 c_8 s_9 \ldots$ be such a sequence.

Lemma 23.1 *The amortized number of faults made by OPT during the phase is at least* $\frac{c}{2}$

Lemma 23.2 *The expected number of faults made by Marking algorithm is at most* cH_k

Proof: $c_1 c_2 c_3$ will lead to eviction of 3 pages from the table. If one of these happens to be s_4 it would be a page fault. Following the line of reasoning we may write probabilities for all s_i's as below:

$$
\begin{aligned}
P[s_4 \text{ leads to a page fault}] &= \frac{3}{100} < \frac{c}{k} \\
P[s_5 \text{ leads to a page fault}] &= \frac{3}{99} < \frac{c}{k-1} \\
&\vdots \\
E[\text{page faults}] = cH_k &\leq c \log k + c
\end{aligned}
\tag{23.1}
$$

■

Thus, the MARKING algorithm has a competitive ratio of $\frac{c \log k + c}{\frac{c}{2}} = 2 \log k + 2$

23.2 Lower bound for randomized algorithm

Lemma 23.3 *No randomized algorithm has competitive ratio better than* $\log k$

Proof:

Overview: We shall prove this by comparing a randomized algorithm, represented by a probability distribution over all deterministic algorithms, with the best deterministic algorithm for worst case probability distribution using Yao's principle. We'll then compute the length of a phase by studying Random walk on a complete graph. Finally, the results are combined to find the lower bound for a randomized algorithms.

Choosing a randomized algorithm: Consider a matrix X with $X(i,j)$ denoting the competitive ratio of deterministic algorithm A_i on sequence σ^j. Randomized algorithm is a probability distribution over all the algorithms. Let P_i be the probability that the randomized algorithm behaves like the algorithm A_i.

$$\begin{aligned} \text{expected c.r. on } \sigma^j &= \sum_i P_i X(i,j) \\ \text{c.r. of randomized } &= \max_j \sum_i P_i X(i,j) \end{aligned} \tag{23.2}$$

The best algorithm would be the one with the best probability distribution which optimizes for competitive ratio.

$$\min_P \max_j \sum_i P_i X(i,j) \tag{23.3}$$

Yao's minmax principle: States that the expected cost of a randomized algorithm on the worst input, is no better than the expected cost for a worst-case probability distribution on the inputs, of the deterministic algorithm that performs best against said distribution.

This implies, considering q to be a probability distribution over σs,

$$\min_P \max_j \sum_i P_i X(i,j) \geq \max_q \min_i \sum_j q_j X(i,j) \tag{23.4}$$

Constructing q: Let S be the set of $k+1$ pages. We construct a probability distribution for choosing a request sequence. The first request r_1 is chosen uniformly from S, the succeeding requests r_{i+1} are chosen uniformly from $S \setminus \{r_i\}$.

Each phase would contain $k+1$ distinct pages and OPT incurs exactly one page fault (why?). Consider any online deterministic algorithm A. The probability of page fault on each request would be $\frac{1}{k}$ as there's only 1 page not in the table. So, the overall cost would be $\frac{1}{k} \times$ length of a phase

Length of a phase: We may map our problem of finding the length of a phase in MARKING to other problems like:

- **Coupon collector's problem:** What is the expected number of boxes to be bought to collect all n coupons given each box contains a random coupon?

- **Random walk:** What is the expected number of steps to visit all vertices in a Random walk.

After visiting the first $1 \dots i$ vertices, our next step would be to one of the remaining $n-1$ vertices, of which $i-1$ are already visited. So,

$$
\begin{aligned}
P[\text{not remaining in } \{1 \dots \text{i}\}] &= \frac{n-i}{n-1} \\
E[\#\text{ steps to visit all vertices}] &= \sum_i \frac{n-1}{n-i} \leq (n-1)\log(n-1) + (n-1)
\end{aligned} \tag{23.5}
$$

We may note that the **length of a phase** $\approx k \log k$.
The overall cost of the deterministic algorithm is $\frac{1}{k} \times k \log k = \log k$

We may now wrap up the proof by plugging in the above result in (23.4); we have the lower bound for the randomized algorithm as $\log k$. ■