CS 591, Lecture 15

Jared Saia
University of New Mexico

Class Outline

- Presentations!!!
- Homework 2
- Finish up MARK Analysis

Class Presentations

- Each student will give a 30 minute presentation
- By October 29th (next Tuesday), you must speak to me (in office hours or by appointment) about your proposed presentation.
- By November 5th, you must turn in a 1 page detailed summary of what you will present. Give a description of the abstract problem, the algorithm(s), and all major theorems.
- Class Presentations will be given on November 19, 21 and 26th - choose a date now.

Class Presentations (II)

- The presentation should be on an algorithmic approach to a "real-world" problem or on a mathematical tool that is useful for real-world algorithms
- There should be some "mathematical meat" to your presentation. In particular, there should be some use of mathematical tools like the ones we learned in class (e.g. graph theory, probability theory, online analysis, etc.)
- It's ok to present in groups of up to 3 people
- Find something that interests you!!!

Ideas for Presentations

- Graph Theory
- Probability Theory
- Spectral Analysis and Search Engines
- Online Algorithms
- Smoothed Analysis
- Computational Biology

Graph Theory

- Data Migration with Space Constraints (see paper “On Algorithms for Efficient Data Migration” on course web page)
- Experimental Results on Data Migration (see paper “An Experimental Study of Data Migration Algorithms” on course page)
- Tutte’s Theorem for matchings in general graphs (see any graph theory book)
- Phylogeny Problems (see Bernard Moret)

Probability Theory

- Pattern Matching and Fingerprinting (see paper “Some applications of Rabin’s fingerprinting method” on course web page)
- How to Reduce the resource bounds for attack-resistant p2p networks (see paper “Censorship Resistant Peer-To-Peer Content Addressable Networks” on course web page)

Spectral Analysis

- The Google Algorithm (see ”The Anatomy of a Large-Scale Hypertextual Web Search Engine” on course web page)
- Theoretical Analysis of why spectral methods work for search (see paper “Spectral Analysis of Data” on course web page)

Online Algorithms

- Online Learning (see paper "Online Algorithms in Machine Learning" on course web page)
- Online Auctions (see me for appropriate references)
- Show that the lower bound for randomized paging against an oblivious adversary is $O(\log n)$

Homework 2

- Problem 4, $n * 2^{-k \log n} = n^{k-1}$. Make sure you simplify all answers as much as possible. Always want a single exponential when possible.
- Problem 5, After compute probability that *a fixed* segment has too many/too few points, need to use union bounds to compute probability that *no* segment has too many/too few points in it.
- Problem 6, don't replace $(1 - 1/n)^n$ with $(1 - 1/e)$ until the very end of the computation. Make sure you *state* that this is an approximation as $n \to \infty$

Homework 2, Problem 7

Part 1)

- Let p be the probability the coin is heads.
- Let a round be two flips of the coin. Say a round succeeds if there is one heads and one tails. Let r be the probability that a round succeeds
- Let t be the expected number of flips until a round succeeds. $t = r * 2 + (1 - r) * (2 + t)$. Simplifying, we get $t = f/r$

HW 2, Problem 7

Part 2)

- We need a more efficient algorithm!
- Hint: Consider the first 4 biased flips
- If the first pair of flips are HT or TH, or the second pair or TH or HT, we can get a bit using the simple method
- Can improve things by pairing up HHTT and TTHH. Idea is to merge the same flips to get a multi-level strategy

Algorithm MARK

We will consider the randomized algorithm MARK defined as follows:

- Initially all pages are marked
- If there is a request for a page p which is in the cache but unmarked, then p is marked
- Otherwise if p is not in the cache, p is brought *marked* into the cache
- If all the pages in the cache are marked when p is brought in, they are all unmarked first
- The page that is evicted is chosen uniformly at random from all unmarked pages currently in the cache.

MARK Analysis

- Defn: $H_k = 1 + 1/2 + 1/3 + \cdots + 1/k$
- Fact: $\ln k < H_k < 1 + \ln k$
- *Theorem:* The competitive ratio of Mark is $2 * H_k$

MARK Analysis

- Fix a request sequence σ and partition it into phases of k unique requests as before
- For each phase i, call the pages in the cache just before the start of the phase the *old* pages and call all other pages requested in the i-th phase *new* pages
- Note that each request in the i-th phase is to either an old or a new phase
- Note also that repeated requests to a new or old page don't contribute to our cost since each page requested is kept in the cache until the end of the phase

MARK Analysis

- Let m_i be the number of new pages requested in the i-th phase
- The worst order for the MARK algorithm is if all requests to new pages precede requests to old pages
- In this order, the requests to new pages incur m_i faults
- What about the remaining $k - m_i$ requests to old pages?

MARK Analysis

- What do we want?: We want to compute the expected number of misses on the $k - m_i$ old pages
- We can get this using linearity of expectation if we can find the right probabilities.
- Q: What is the probability that the j-th old page requested will not be in the cache? (i.e. it will generate a fault)

Mark Analysis

- Q: What is the probability that the j-th old page requested is not in the cache?
- Let x_j be the j-th old page
- Let O be the set of old pages.
- Let $O_j = O - \{x_1, \ldots, x_{j-1}\}$ (i.e. the set of unmarked old pages when x_j is requested)
- Note that m_i of the pages in O_j have been evicted from the cache when x_j is requested. These m_i evicted pages are chosen uniformly at random from the set O_j
- Thus the probability that some fixed page (say x_j) in the set O_j has been evicted is $m_i/|O_j| = m_i/k - (j-1)$
- Thus the probability that the j-th old page requested is not in the cache is $m_i/k - j + 1$

Mark Analysis

Hence the number of faults incurred by MARK is:

$$
\begin{aligned}
m_i + \sum_{j=1}^{k-m_i} \frac{m_i}{k-j+1} &= m_i(H_k - H_{m_i} + 1) && (1)\\
&\leq m_i H_k && (2)
\end{aligned}
$$

Bound for OPT

It's not hard to bound OPT for this sequence:

- Let q_i be the number of faults to OPT in the i-th phase and let l be the total number of phases
- For $i > 2$, $q_{i-1} + q_i \geq k + m_i$, since $k + m_i$ distinct pages requested during these two phases.

$$
\begin{aligned}
\sum_{i=1}^{l} (q_{i-1} + q_i) &\geq \sum_{i=2}^{l} m_i && (3)\\
2 * \sum_{i=1}^{l} q_i - q_1 &\geq \sum_{i=2}^{l} m_i && (4)\\
2 * OPT &\geq q_1 + \sum_{i=2}^{l} m_i && (5)\\
2 * OPT &\geq \sum_{i=1}^{l} m_i && (6)
\end{aligned}
$$

Take Away

- So MARK has no more than $H_k * \sum_i m_i$ faults
- OPT has at least 1/2($\sum_i m_i$) faults
- Hence the competitive ration is $2 * H_k$
- Note: There are variants of MARK that have been shown to be H_k competitive