CS 388R: Randomized Algorithms, Fall 2023 9/6/2023

Lecture 5: Treaps, Coupon Collector, Balls and Bins

Prof. Eric Price *Scribe: Gary Wang, Pranav Venkatesh*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

In last lecture we covered Game Tree Evaluation.

In this lecture, we are going to explore 3 interesting problems:

- Treaps
- Balls and Bins
- Coupon Collector Problem

2 Treaps

Problem Definition: We must construct a randomized data structure with the properties of a binary search tree and heap.

Construction: First, we assign a random weight to each element. In a recursive manner, we pick the smallest weight as the root and propagate nodes to the left or right subtree based on their random weight.

Operations: Each insert and remove operation on the treap must preserve the weighted structure. The treap supports dynamic operations, meaning that the state is a randomly constructed BST at all times.

Does this remind you of anything else? Quicksort! We similarly pick a random element and split into left and right partitions.

We know that the runtime of quicksort is $\Sigma_{x \in T} depth(x)$, meaning an average time complexity of $O(n \log n)$.

Maximum Depth Analysis: We must show that the maximum depth is $O(\log n)$ with high probability $\implies$ Quick sort is $O(n \log n)$. This analysis will be rather simple and not so tight.

We will be able to show that the depth, with high probability, is $1 - \frac{1}{n^c}$, where c is a constant!

Let us define d(x) as the depth of some element x and l as the “layer”.

$Pr[\max d(x) \geq l] \leq n - \max_{x \in X} Pr[d(x) \geq l]$

It suffices to show that depth of x for all x is $O(\log n)$. Let x be random element. We first start out with $k_1 = n$ in x's subtree at layer 1. After picking an element: k_i elements in x's subtree at layer i. Let $k_i = 0$ if x is at layer before i. $d(x)$ is max i such that $k_i \geq 1$. We know, therefore, that:

$Pr[d(x) \geq l] = Pr[k_l \geq 1]$ (at layer l there is at least 1 element)

Can we show that k_l is large with small probability?

$k_1 = n$

$Pr[k_2 < \frac{3}{4}k_1] \geq \frac{1}{2}$, regardless of x

If partitioned element is between the first and third quartile elements it always works, and the probability of having that is $\frac{1}{2}$.

For all i, $Pr[k_i \leq \frac{3}{4}k_{i-1}] \geq \frac{1}{2}$, regardless of choices made in ALL previous rounds.

Define z_i to be 1 if $k_i \leq \frac{3}{4}k_i$ for all i and 0 otherwise.

$Pr[z_i] \geq \frac{1}{2}$ (same conditioned on all previous z)

$Pr[k_l \geq 1] \geq Pr[\Sigma_{i=1}^{l} z_i \leq \log_{\frac{4}{3}} n]$

Chernoff Bound: We may now attempt to use a Chernoff Bound. We know that the expected sum is at least $\frac{l}{2}$.

$Pr[\Sigma_{i=1}^{l} z_i \leq E - (\frac{l}{2} - \log_{\frac{4}{3}} n)] \leq exp(-\frac{2(\frac{l}{2} - \log_{\frac{4}{3}} n)^2}{l}) \implies$ If l is big (greater than $8c \log n$, this value becomes $e^{-\frac{l}{8}}$ and probability of failure is n^{-c}).

We may conclude that the depth, therefore, is order of $\log n$.

Can we really conclude this though? We have a “small” issue. We can only apply Chernoff Bound on events that are independent. However, z events are not independent $\rightarrow$ how do we solve this?

This statement is independent: $Pr[z_i] \geq \frac{1}{2}$ (same conditioned on all previous z); The one half is guaranteed no matter what happens prior.

Possible Solutions:

1. Find a statement of Chernoff that handles it! (Consult literature)

2. Use Azuma's Inequality (involves martingales): Left as exercise to reader (go on wikipedia)

3. Use Stochastic Domination

Ex: Stochastic Domination

Given all z variables, $Pr[z_i|\text{previous } z's] \geq \frac{1}{2}$

There exists variables y coupled to z, joint distribution, such that:
$y_i < z_i$ and $Pr[y_i|\text{previous } y's] = \frac{1}{2}$

The y variables are independent and therefore Chernoff bound applies to y_i.

Additionally, the sum probability of z is less than sum probability of y, and therefore the original conclusion holds.

3 Coupon Collector

Problem Statement: There are n distinct Pokemon cards. There are cereal boxes that come with a random Pokemon card. How many cereal boxes does one need to buy to "catch them all"?

T_i = time it takes to get the i^{th} new item

Expected Value: We know that $E[T_1] = 1$, $E[T_n] = n$. At the i^{th} item there are $(n+1-i)$ good items, meaning:

$E[T_i] = \frac{n}{n+1-i}$

$E[\Sigma T_i] = n(\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} \cdots + 1) = n \cdot H_n = \Theta(n \log n)$

We will revisit this problem later!

4 Balls and Bins

Problem Statement: We randomly put n balls into n bins: what happens? What are some properties about how the balls are distributed across the bins?

Some questions to address:

1. What is the max load of a bin with high probability?
2. What is the average load over balls?

Question 1: Max Load

Max load is at most n (obviously)

What about with high probability?

Union bound max load: $Pr[\max x_i \geq l] \leq n \cdot \mathbb{P}[x_i \geq l]$

Additive Chernoff bound:

$z_i = 1$ if ball i lands in bin 1.

$x_1 = \Sigma z$

$Pr[x_1 \geq 1 + t] \leq e^{-\frac{2t^2}{n}}$, t = $\sqrt{n \log n}$ with high probability

Multiplicative Chernoff bound:

$Pr[x_1 \geq (1 + t)l] \leq e^{-\frac{t^2}{2+t}}$, $e^{-\frac{t}{2}} \leq n^{-c}$ for $t = O(\log n)$

Bennett's inequality can give a better bound!

Direct calculation:

$Pr[x_1 \geq l] \leq \binom{n}{l}\frac{1}{n^l}$

Bound binomial coeff: $(\frac{n}{k})^k \leq \binom{n}{k} \leq (\frac{en}{k})^k$

$Pr[x_1 \geq l] \leq (\frac{en}{l})^l \frac{1}{n^l} = (\frac{e}{l})^l$

$Pr[x_i \geq l] \leq n \cdot (\frac{e}{l})^l$

$(\frac{e}{l})^l \leq n^{-c}$

$l \log(\frac{l}{e}) = c \log n$

$l = \frac{\log n}{\log \log n}$

LHS $= \frac{A \log n}{\log \log n}(\log \log n - \log \log \log n + \log \frac{A}{e}) = \Theta(logn)$ ... *black magic*

$\max x_i = O(\frac{\log n}{\log \log n})$ with high probability.

We will explore problem 2 next lecture!

References

[AMS99] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.