# Assignment 1 — Applied Algorithms, T. II/2024–25

Austin Jetrin Maddison 6481268

February 8, 2025

## Problem 1. Las Vegas and Monte Carlo

a.i) We want to show the probablity of running time of Monte Carlo is atleast the worst running time which is $4f(n)$. We can use markov inqualities to bound it..

$$\mathbf{P}(X \leq \lambda) \leq \frac{E[X]}{\lambda}$$

$$\begin{aligned} \mathbf{P}(T(n) \leq 4f(n)) &\leq \frac{f(n)}{4f(n)} \\ &\leq \frac{1}{4} \end{aligned}$$

a.ii) The worse-case running time happens at most 1/4 which produces incorrect answers. We can get the complement of the last answer...

$$1 - \mathbf{P}(T(n) \leq 4f(n)) \leq 1 - \frac{1}{4} = \frac{3}{4}$$

b.i) The LV algorithm running time is described as the follwing. Each iteration requires running A to produce an answer then run C to check the answer. So the running time for each trial is...

$$1 \text{ iteration running time of LV} = f(n) + g(n)$$

So the question is what is the expected iterations needed to run LV to get a correct answer. If p is the probablity of success then the expected $1/p$.

$$\text{Running time of LV} = \frac{1}{p}(f(n) + g(n))$$

## Problem 2. Chernoff-Hoeffding With Bounds

2.1)

$$\Pr[X > (1+\beta)\mu] \leq \exp\left(-\frac{\beta^2}{2+\beta}\mu H\right)$$
$$\Pr[X > (1+\varepsilon)\mu H] \leq \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu H\right)$$

$$(1+\epsilon)\mu_H = (1+\beta)\mu$$
$$\frac{\mu_H}{\mu} = \frac{1+\epsilon}{1+\beta}$$

2.2)

$$\Pr[X > (1+\beta)\mu] \leq \exp\left(-\frac{\beta^2}{2+\beta}\mu H\right)$$
$$\Pr[X > (1+\varepsilon)\mu H] \leq \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu H\right)$$

$$(1+\epsilon)\mu_H = (1+\beta)\mu$$
$$\frac{\mu_H}{\mu} = \frac{1+\epsilon}{1+\beta}$$

2.3)

$$
\begin{aligned}
\Pr[X > (1+\beta)\mu] &\leq \exp\left(-\frac{\beta^2}{2+\beta}\mu H\right) \\
\Pr[X > (1+\varepsilon)\mu H] &\leq \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu H\right)
\end{aligned}
$$

$$
\begin{aligned}
(1+\epsilon)\mu_H &= (1+\beta)\mu \\
\frac{\mu_H}{\mu} &= \frac{1+\epsilon}{1+\beta}
\end{aligned}
$$

2.4)

$$
\begin{aligned}
\Pr[X > (1+\beta)\mu] &\leq \exp\left(-\frac{\beta^2}{2+\beta}\mu H\right) \\
\Pr[X > (1+\varepsilon)\mu H] &\leq \exp\left(-\frac{\varepsilon^2}{2+\varepsilon}\mu H\right)
\end{aligned}
$$

$$
\begin{aligned}
(1+\epsilon)\mu_H &= (1+\beta)\mu \\
\frac{\mu_H}{\mu} &= \frac{1+\epsilon}{1+\beta}
\end{aligned}
$$

## Problem 3. Rescaling Trick

(Statement of problem goes here.)

*Proof.* (Type your proof here.) $\square$

## Problem 4. $x^2$ With $\pi$ Degrees of Freedom

(Statement of problem goes here.)

*Proof.* (Type your proof here.) $\square$

## Problem 5. Simple Samplers.

(Statement of problem goes here.)

*Proof.* (Type your proof here.) $\square$

## Problem 6. Median of Means

(Statement of problem goes here.)

*Proof.* (Type your proof here.) $\square$

## Problem 7. Skip List

### Experimental Setup

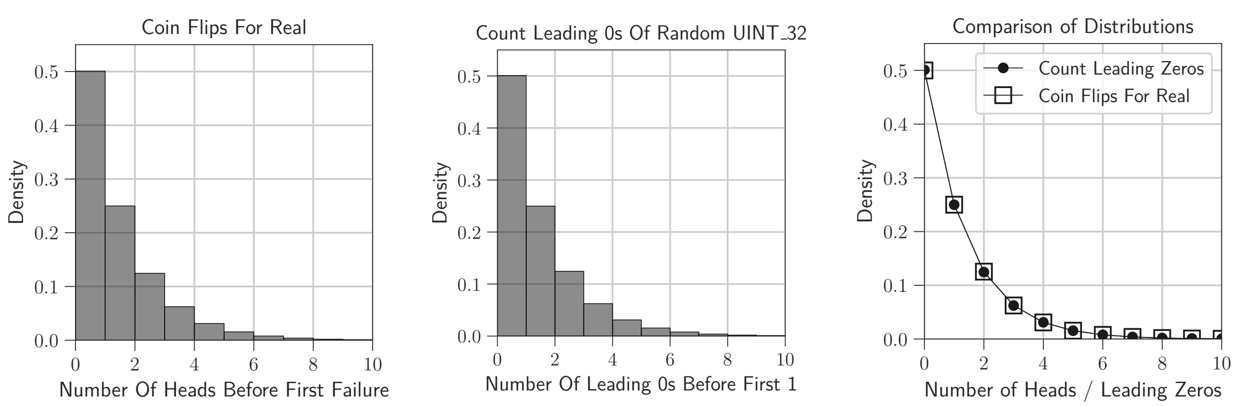
Conducted our benchmarks using Google Benchmark on a system with the following hardware specifications:

- **Processor**: 11th Gen Intel(R) Core(TM) i7-11370H 8-core CPU @ 3.3 GHz
- **Cache Hierarchy**:
  - L1 Data: 48 KiB per core (×4)
  - L1 Instruction: 32 KiB per core (×4)
  - L2 Unified: 1280 KiB per core (×4)
  - L3 Unified: 12,288 KiB (shared)

All benchmarks were compiled using `-Ox` optimization level with `MSVC` (version 19.42.34436) and executed in a single-threaded environment to minimize external interference.

### Questions And Experiments

Q1.) Can we perform count coin tosses differently and is the alternative better?




| Benchmark | Time (ns) | CPU (ns) | Iterations |
|---|---|---|---|
| Coin Flip For Real | 29.8 | 29.3 | 22,400,000 |
| Coin Flip Count Leading 0s | 5.11 | 4.87 | 144,516,129 |

Table 1: Benchmark results comparing different coin flip implementations.

Q2.) How does varying max height change performance?
Q3.) Linked lists are known to be cache unfriendly, is there a way we can modify

Q4.) How does it perform against a reputable ordered map?

b.) Search algorithm of Skip List when start is at the bottom left corner in $O(\log(d))$ where d is the number of elements smaller than the key?

## Problem 8. $(a, b)$ tree. $(2, 3)$ tree.

a.)

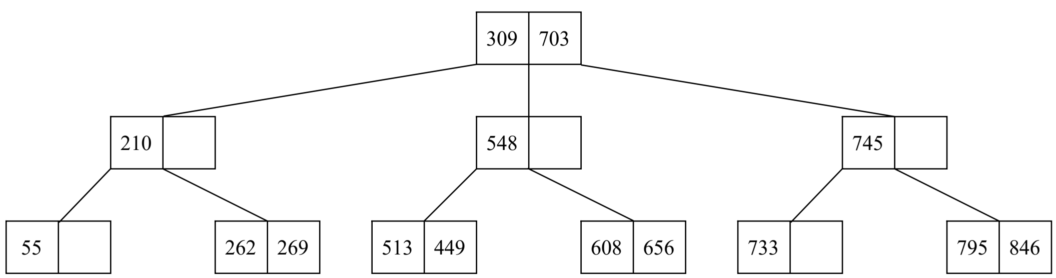



Figure 1: Keys $733, 703, 608, 846, 309, 269, 55, 745, 548, 449, 513, 210, 795, 656, 262$ inserted into a $(2, 3)$ tree.

b.)

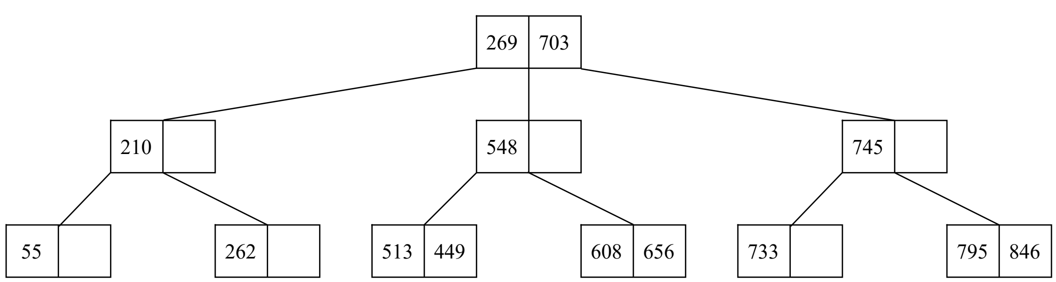



Figure 2: Key 309 removed from Figure 1 tree.

## Problem 9. B-Tree Speed

(Statement of problem goes here.)

*Proof.* (Type your proof here.) $\square$