

Problem Set 2

Spring 2025

Due: 11:59 pm Monday, 3 Mar

To facilitate grading and timely feedback
please note that all submissions are through Gradescope.

Solve each problem on a new page and put your name on each page.
Clearly identify all collaborations and resources used in preparing the solutions.
You should typeset your solutions using \LaTeX and correctly link
solutions on Gradescope.

1. Here is another way to analyze randomized quicksort. Let X_{ij} be the indicator random variable given by

$$X_{ij} = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ smallest item in the list ever gets compared with the } j^{\text{th}} \text{ smallest one} \\ 0 & \text{otherwise.} \end{cases}$$

The total number of comparisons then is

$$X = \sum_{1 \leq i < j \leq n} X_{ij}.$$

Show that this gives

$$\mathbb{E}[X] = 2 \sum_{j=1}^n (H_j - 1),$$

where $H_j = \sum_{k=1}^j \frac{1}{k}$ is the j^{th} harmonic number.

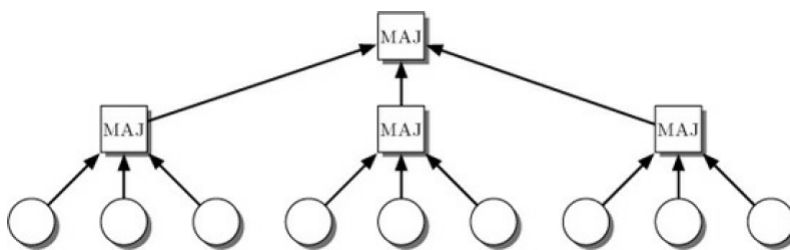


Figure 1: Majority tree of depth 2.

2. Suppose we have a tree where each node has three children, so a tree of depth k has $N = 3^k$ leaves. Place a *MAJORITY* gate at each node which outputs the majority of its children's truth values. Show that by evaluating the subtrees in random order, we can evaluate the entire tree while only looking at

$$\left(\frac{8}{3}\right)^k = N^{\log_3(8/3)}$$

leaves on average.

3. We are given a coin that may or may not be biased, and we would like to compute an accurate estimate of the probability of heads. Specifically, if the actual unknown probability of heads is p , we would like to compute an estimate \tilde{p} such that

$$\Pr(|\tilde{p} - p| > \epsilon) < \delta,$$

where ϵ is a given accuracy or error parameter, and δ is a given confidence parameter. The following algorithm is a natural first attempt. Here *Flip()* returns the result of an independent flip of the unknown coin.

Algorithm 1 Algorithm to estimate mean

```

procedure MEANESTIMATE( $\epsilon$ )
     $count \leftarrow 0$ 
    for  $i \leftarrow 1$  to  $N$  do
        if  $Flip() = Heads$  then
             $count \leftarrow count + 1$ 
        end if
    end for
    return  $count/N$ 
end procedure

```

- (a) Let \tilde{p} denote the estimate returned by *MeanEstimate*(ϵ). Prove that $\mathbb{E}(\tilde{p}) = p$.
- (b) Prove using Chebyshev's inequality that if we set $N = \lceil \frac{\alpha}{\epsilon^2} \rceil$ for some appropriate constant α , then we have

$$\Pr(|\tilde{p} - p| > \epsilon) < \frac{1}{4}.$$

4. Suppose we have a binary counter such that the cost to increment or decrement the counter is equal to the number of bits that need to be flipped. We saw in class that if the counter begins at 0, and we perform n increments, the amortized cost per increment is just $O(1)$. Equivalently, the total cost to perform all n increments is $O(n)$.

Suppose that we want to be able to both increment and decrement the counter.

- (a) Show a sequence of n operations allowing both increments and decrements and starting from 0 that, without ever making the counter go negative, costs $\Omega(\log n)$ amortized per operation (i.e., $\Omega(n \log n)$ total cost). To reduce the cost observed in part (a) we'll consider the following *redundant ternary number system*. A number is represented by a sequence of *trits*, each of which is 0, +1, or -1. The value of the number represented by t_{k-1}, \dots, t_0 (where each $t_i, 0 \leq i \leq k-1$ is a trit) is defined to be

$$\sum_{i=0}^{k-1} t_i 2^i.$$

For example, $\boxed{1} \boxed{0} \boxed{-1}$ is a representation for $2^2 - 2^0 = 3$.

The process of incrementing a ternary number is analogous to that operation on binary numbers. You add 1 to the low order trit. If the result is 2, then it is changed to 0, and

a carry is propagated to the next trit. This process is repeated until no carry results. Decrementing a number is similar. You subtract 1 from the low order trit. If it becomes -2 then it is replaced by 0, and a borrow is propagated. Note that the same number may have multiple representations (e.g., $\boxed{1}\boxed{0}\boxed{1} = \boxed{1}\boxed{1}\boxed{-1}$). That is why this is called a redundant ternary number system. The cost of an increment or a decrement is the number of trits that change in the process.

- (b) Starting from 0, a sequence of n increments and decrements is done. Give a clear, coherent proof that with this representation, the amortized cost per operation is $O(1)$ (i.e., the total cost for the n operations is $O(n)$). Hint: think about a “bank account” or “potential function” argument.
5. Tabulated hashing uses tables of random numbers to compute hash values. Suppose $|\mathcal{U}| = 2^w \times 2^w$ and $m = 2^l$, so the items being hashed are pairs of w -bit strings and hash values are l -bit strings. Let $A[0 \dots 2^w - 1]$ and $B[0 \dots 2^w - 1]$ be arrays of independent random l -bit strings, and define the hash function $h_{A,B} : \mathcal{U} \mapsto [m]$ by setting

$$h_{A,B}(x, y) = A[x] \oplus B[y]$$

where \oplus denotes bit-wise exclusive-or. Let \mathcal{H} denote the set of all possible functions $h_{A,B}$. Filling the arrays A and B with independent random bits is equivalent to choosing a hash function $h_{A,B} \in \mathcal{H}$ uniformly at random.

- (a) Prove that \mathcal{H} is 2-uniform.
- (b) Prove that \mathcal{H} is 3-uniform.