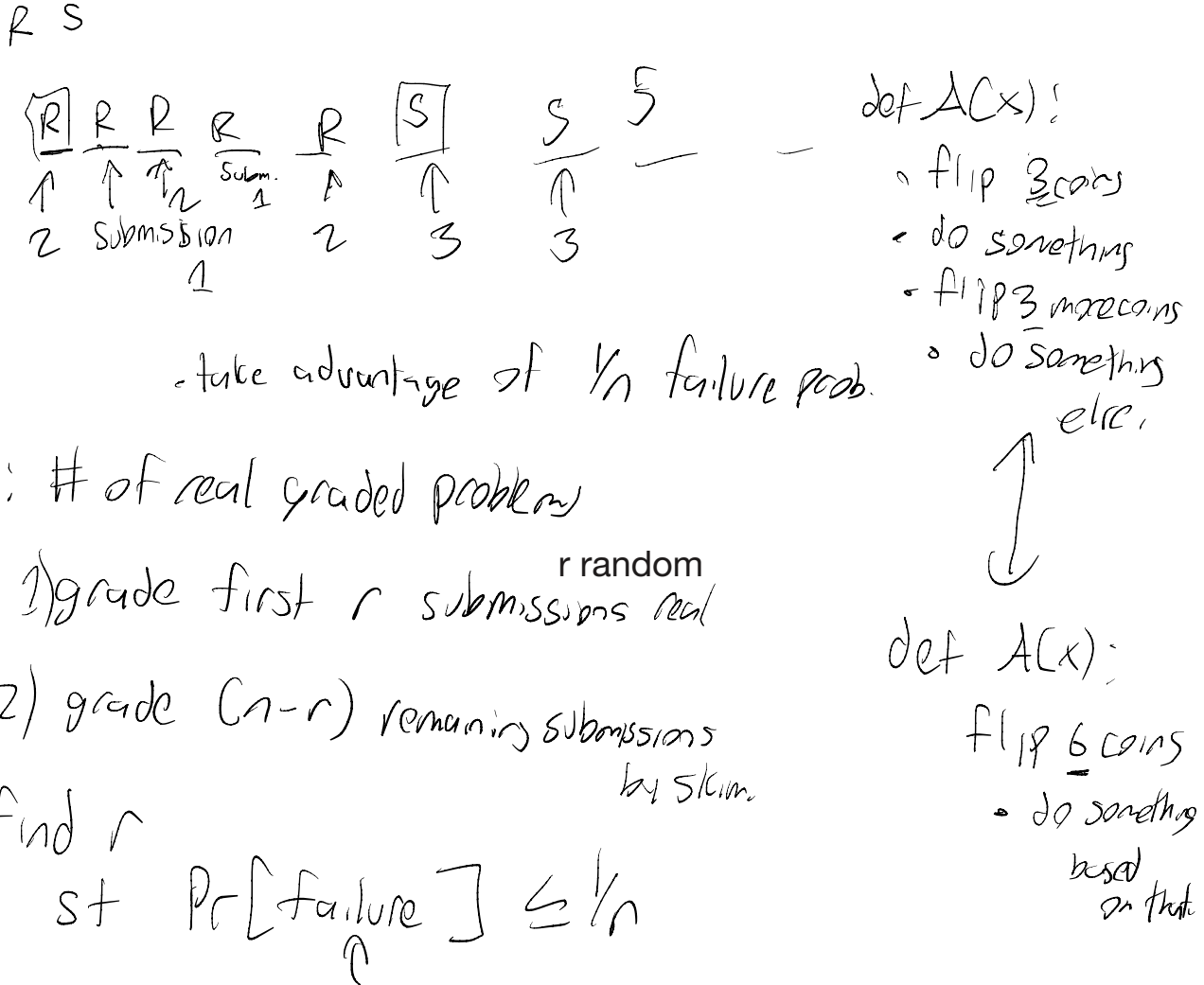
Dishan's Thanksgiving Problem The final results are back, and the 381 GTAs need to grade a whopping n student submissions! They want to spend as little time as possible grading (they want to play GTA 6). They can choose between two methods for grading any given submission:

- **Real grading:** This always takes a couple of painstaking minutes per submission, but it guarantees an accurate grade. The solution type will also be cached into the GTAs brain.
- **Skimming:** This method takes negligible time. However, the grader will skim so quickly that they only have a 50% chance of giving an accurate grade! That is, of course, unless they've seen the solution type before (in which case it is guaranteed with 100% chance to be accurate).

Unfortunately, the students have many different solution types. A preliminary ChatGPT analysis showed that every solution type had at least k instances among the n submissions.

Give a randomized algorithm to minimize the total number of submissions that were *real graded* such that the probability of grading any submission inaccurately is at most $1/n$ (the GTAs can handle a couple regrade requests, after all). Express the asymptotic bound of real graded submissions in terms of n and k.



$$\Pr[\exists\, i \in [K]:\ \text{problem type } i \text{ graded incorrectly}]$$

$$= \Pr[\text{type 1 incorrect or type 2 incorrect or } \ldots \text{ type } K \text{ incorrect}]$$

$$\leq \sum_{i=1}^{n/K} \Pr[\text{type } i \text{ incorrect}] \quad \text{(union bound)}$$

$$= \binom{n}{K} \Pr[\text{fixed type } i \text{ incorrect}]$$

$$= \Pr[\text{haven't seen type } i \text{ in the } r \text{ real subm.}]$$

$$\leq \binom{n}{K}\left(\frac{n-K}{n}\right)^r \leq 1/n$$

goal: find bound on r $\qquad (1-x)^y \leq e^{-xy}$

$$\binom{n}{K}(1-K/n)^r$$

$$\leq \binom{n}{K} e^{-Kr/n} \leq 1/n$$

$$e^{-Kr/n} \leq K/n^2$$

$$-kr/n \leq \log k/n^2$$

$$\frac{kr}{n} \geq \log\left(\left(k/n^2\right)^{-1}\right)$$

$$\frac{kr}{n} \geq \log\left(n^2/k\right)$$

$$r \geq \frac{n}{k}\log\left(n^2/k\right)$$

$$n \text{ submissions} = \left(\begin{array}{c}\text{instances} \\ \text{of each} \\ \text{type}\end{array}\right) \times \left(\#\text{problem types}\right)$$

$$n/k \geq \frac{n \text{ submissions}}{\#\text{instances per type}} = \#\text{problem types}$$

Turing Machine. Recall that a turing machine $\mathcal{M}$ is represented by the 7-tuple $(Q, \Sigma, \tau, \delta, q_0, q_{acc}, q_{rej})$. The language of $\mathcal{M}$ is all the strings which lead to acceptance, $L(\mathcal{M}) = \{x \in \Sigma^* : q_{accept} \leftarrow \mathcal{M}(x)\}$. Note that a turing machine may not halt on strings x which it does not accept.

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two turing machines. Show how we can construct:

1. $L(\mathcal{M}_1 \cap \mathcal{M}_2) := L(\mathcal{M}_1) \cap L(\mathcal{M}_2)$;
2. For any fixed $k \in \mathbb{N} : L_k(\mathcal{M}_1) := \{x \in L(\mathcal{M}_1) : \mathcal{M}_1(x) \text{ accepts in } k \text{ steps}\}$
3. $L(\mathcal{M}_1 \cup \mathcal{M}_2) := L(\mathcal{M}_1) \cup L(\mathcal{M}_2)$;

2^n

Prop: Turing machines $\text{poly}(T(n))$ $\text{poly}(n \log n)^{381}$

$\updownarrow$ $\updownarrow$

Python code. $T(n)$ $(n \log n)^{381}$

- Turing machines $\mathcal{M}$ accept & reject

$\updownarrow$

$L(\mathcal{M}) = \{x : \mathcal{M}(x) \text{ accept}\}$

- $\mathcal{M}$ might infinite loop.

Q : set of states

Σ : alphabet $= \{0, 1\}$

δ : transition function $= \delta : Q \times \Sigma$

$\hookrightarrow \{R, W\} \times \{L, R\}$

$q_0 \in Q$: starting state