

CPSC 421: Introduction to Theory of Computing
Practice Problem Set #6, Not to be handed in

Note: $\mathbb{N} = \{0, 1, \dots\} \subset \{1, 2, \dots\} = \mathbb{N}_+$. For simplicity (and wlog), the alphabet is $\{0, 1\}$, unless stated otherwise.

1. (Easy) Recall the definition of RP: $L \in \text{RP}$ if there exists a polytime PTM M such that

$$\begin{aligned}x \in L &\Rightarrow \mathbb{P}(M(x) = 1) \geq 1/2 \\x \notin L &\Rightarrow \mathbb{P}(M(x) = 0) = 1.\end{aligned}$$

Thus, M errs only when $x \in L$, and the probability of error is at most $1/2$. Now, run M twice on x and accept if at least one of the runs of M on x accepts. *Compute the new error probability and conclude that $\text{RP} \subset \text{BPP}$.*

2. (Medium) Consider the following two definitions of RP:

Definition 1. (RP1): $L \in \text{RP1}$ if there exist a polytime PTM M and a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$, where $p \geq 1$, such that for every $x \in \{0, 1\}^*$

$$\begin{aligned}x \in L &\Rightarrow \mathbb{P}(M(x) = 1) \geq \frac{1}{p(|x|)} \\x \notin L &\Rightarrow \mathbb{P}(M(x) = 0) = 1.\end{aligned}$$

Definition 2. (RP2): $L \in \text{RP2}$ if there exist a polytime PTM M and a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$, where $p \geq 1$, such that for every $x \in \{0, 1\}^*$

$$\begin{aligned}x \in L &\Rightarrow \mathbb{P}(M(x) = 1) \geq 1 - 2^{-p(|x|)} \\x \notin L &\Rightarrow \mathbb{P}(M(x) = 0) = 1.\end{aligned}$$

Show that $\text{RP1} = \text{RP2}$.

3. Suppose you download a large movie from an internet server. Before watching it, you'd like to check that your downloaded file has no errors; i.e., the file on your machine is *identical* to the file on the server. You would like to do this check without much additional communication, so sending the entire file back to the server is not a good solution. Ignoring cryptographic considerations, this is essentially the problem of computing a checksum and there are standard ways to do this; e.g., CRCs.

For concreteness, say that the file is n bits long, the server has the bitvector $a = (a_1, \dots, a_n)$ and you have the bits $b = (b_1, \dots, b_n)$.

We'd like a guarantee of this sort:

- For every vectors a and b , our algorithm will flip some random coins, and for most outcomes of the coins, will detect whether or not a and b are identical.

Define polynomials $f_a(x) = \sum_{i=1}^n a_i x^i$ and $f_b(x) = \sum_{i=1}^n b_i x^i$. We will view these as polynomials over a field \mathbb{F}_p , where p is a prime number; in other words, think of \mathbb{F}_p as the set of numbers $\{0, \dots, p-1\}$, and when we evaluate the polynomials at some point x , compute the answer modulo p . Now define $g = f_a - f_b$.

- (a) (Easy with the hint) Fix a prime number, p . Give an upper bound on the probability that a uniformly chosen $x \in \mathbb{F}_p$ is a root of g . **Hint:** you may use the following **Theorem:** Let $f(x)$ be a non-zero polynomial of degree at most d in a single variable x over any field. Then f has at most d roots (i.e., f evaluates to zero on at most d elements of the field).
- (b) Consider the following algorithm. You and the server agree on the prime p . The server picks $x \in \mathbb{F}_p$ uniformly at random. It sends you x and $f_a(x)$. You compute $g(x) = f_a(x) - f_b(x)$. If $g(x) = 0$ the algorithm announces “ a and b are equal”. If $g(x) \neq 0$ the algorithm announces “ a and b are not equal”.
- (Easy with the hint) What is the computational complexity of picking the prime p , and how many bits are required to transmit x and $f_a(x)$? **Hint:** A fact known as *Bertrand’s Postulate* implies that for any $n \in \mathbb{N}_+$, there always exists a prime in $[2n, 4n]$.
 - (Easy) Is the algorithm in the previous part one-sided or two-sided error? Explain your answer.
4. (Hard without hint; Medium with hint (requires work)) **The weakest possible BPP definition.** Show that $L \in \text{BPP}$ if and only if there exist a polynomial-time computable function $f : \mathbb{N} \rightarrow [0, 1]$, a positive polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$, and a polytime PTM M such that for every $x \in \{0, 1\}^*$:

$$x \in L \Rightarrow \mathbb{P}(M(x) = 1) \geq f(|x|) + \frac{1}{p(|x|)}$$

$$x \notin L \Rightarrow \mathbb{P}(M(x) = 1) < f(|x|) - \frac{1}{p(|x|)}.$$

Hint: On input x , define a new PTM, N , that invokes $M(x)$ n times for some n to be determined. Compute $\hat{p} = \frac{1}{n} \sum_{i=1}^n t_i$, where $t_i \in \{0, 1\}$ is the result of the i th invocation of M on x . Accept if $\hat{p} > f(|x|)$ and otherwise reject. Now apply a version of [Chernoff’s inequality](#).

5. (Easy if you solved previous problem; Medium if using hint from previous problem; otherwise Hard) **The strongest possible BPP definition.** Show that for every $L \in \text{BPP}$ and every positive polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$, there is polytime PTM M such that for every $x \in \{0, 1\}^*$:

$$x \in L \Rightarrow \mathbb{P}(M(x) = 1) \geq 1 - 2^{-p(|x|)}$$

$$x \notin L \Rightarrow \mathbb{P}(M(x) = 1) < 2^{-p(|x|)}.$$