We recall the definition of \( \text{BPP} \). A language \( L \) is in \( \text{BPP} \) if there exists a Turing machine \( M \) running in polynomial time \( p(n) \) on all input \( x \) such that \( |x| = n \) and random tape \( r \) of size \( p(n) \) such that:

- If \( x \in L \), then \( \Pr_{r}[M(x, r) = \top] \geq 2/3 \);
- If \( x \notin L \), then \( \Pr_{r}[M(x, r) = \top] \leq 1/3 \).

**Exercise 1** (\( \text{RP}^* \)). We define \( \text{RP}^* \) as the class of all languages \( L \) for which there exists a probabilistic Turing machine \( M \) running in polynomial time, such that:

- If \( x \in L \) then \( \Pr[M(x, r) \text{ reject}] < 1 \)
- If \( x \notin L \) then \( \Pr[M(x, r) \text{ accept}] = 0 \)

Do you recognize this class?

**Exercise 2** (A quick come back to oracles). Give an lower and upper bound on the following complexity classes: \( \text{NP}^{\text{NP}}, \text{RP}^{\text{RP}} \) and \( \text{BPP}^{\text{BPP}} \).

**Exercise 3** (\( \text{NP} \), \( \text{RP} \) and \( \text{BPP} \)). Show that if \( \text{NP} \subseteq \text{BPP} \) then \( \text{NP} = \text{RP} \).

**Exercise 4** (Logarithmic space \( \text{BPP} \)). Define \( \text{BPL} \) as the class of languages decided by a probabilistic Turing machine running in logarithmic space and polynomial time (with a \( \text{BPP} \)-like semantic). Show that \( \text{BPL} \subseteq \text{P} \).

**Exercise 5** (The \( \text{PP} \) class). The class \( \text{PP} \) is the class of languages \( L \) for which there exists a polynomial time probabilistic Turing machine \( M \) such that:

- if \( x \in L \) then \( \Pr[M(x, r) \text{ accepts}] > \frac{1}{2} \)
- if \( x \notin L \) then \( \Pr[M(x, r) \text{ accepts}] \leq \frac{1}{2} \)

1. Show that \( \text{BPP} \subseteq \text{PP} \) and \( \text{NP} \subseteq \text{PP} \);
2. Exhibit a \( \text{PP} \)-complete language;
3. Show that \( \text{PP} \) is closed under complement;
4. Consider the decision problem \( \text{MAJSAT} \):
   
   (a) Input: a boolean formula \( \phi \) on \( n \) variables
   (b) Output: the (strict) majority of the \( 2^n \) valuations satisfy \( \phi \).
Show that MAJSAT ∈ PP. In fact, MAJSAT is PP-complete.

One may also consider the decision problem MAXSAT:

(a) Input: a boolean formula φ on n variables, a number K
(b) Output: more than K valuations satisfy φ.

Show that MAXSAT is also PP-complete (to prove that MAXSAT ∈ PP one may reduce MAXSAT to MAJSAT).

Exercise 6 (A little come back to P and RP). We define a random language A by setting that each word $x \in \{0, 1\}^*$ is in A with probability $1/2$. Show that almost surely (on the probabilistic choice on the language A) we have $P^A = RP^A$.

Hint: Fix an $\epsilon > 0$ and an enumeration $(M_i)_{i \in \mathbb{N}}$ of probabilistic Turing machine running in polynomial time with an oracle. Exhibit deterministic polynomial time Turing machines $(N_i)_{i \in \mathbb{N}}$ and consider the probability (over the random language considered) that there is one $i$ such that $M_i$ and $N_i$ does not coincide.