**Exercise 1.**
Let us define BPL as $\text{BPSPACE}(\log(n))$, and $\text{RL} = \text{RSPACE}(\log(n))$.

1. Show that $\text{RL} \subseteq \text{BPL}$.
2. Show that $\text{BPL} \subseteq \text{P}$. (*Hint: Consider the ‘adjacency matrix’ of the configuration graph.*)
3. Show that $\text{BPL} \subseteq \text{SPACE}(O(\log^2 n))$.

**Exercise 2.**

**Definition.** A problem with promise $\Pi$ is two disjoint sets $\Pi_Y$ and $\Pi_N$. On input $x$, where its belonging to $\Pi_Y \cup \Pi_N$, the problem is to decide whether $x \in \Pi_Y$ or $x \in \Pi_N$.

$\Pi \in \text{PromiseBPP}$ if there exists a PPT Turing machine $M$ such that if $\Pr[M(x) = 1 \mid x \in \Pi_Y] \geq 2/3$ and $\Pr[M(x) = 1 \mid x \in \Pi_N] \leq 1/3$.

Let $\text{AEA}$ (Additive Error Acceptance) be the problem defined by $\text{AEA}_Y = \{C \mid \Pr_r[C(r) = 1] \geq 2/3\}$ and $\text{AEA}_N = \{C \mid \Pr_r[C(r) = 1] \leq 1/3\}$.

1. Show that $\text{BPP} \subseteq \text{PromiseBPP}$.
2. Show that $\text{AEA}$ is PromiseBPP-complete (under Karp-reduction).

Let $\epsilon \leq 1/6$, an $\epsilon$-PCA is the problem with promise given by $\epsilon$-PCA$_Y = \{(C, p) \mid \Pr_r[C(r) = 1] \geq p + \epsilon\}$ and $\epsilon$-PCA$_N = \{(C, p) \mid \Pr_r[C(r) = 1] \leq p\}$.

1. Show that $\epsilon$-PCA is PromiseBPP-complete.

PromiseBQP is defined as the class of problems ($\Pi_Y, \Pi_N$) that can be decided by a uniform family of quantum circuits: a uniform family of circuits $Y$, acting on $\text{poly}(n)$ qubits, decides if a string $x$ of length $n$ is a YES-instance or NO-instance with probability at least 2/3.

4. Show that $q\text{AEA}_Y = \{C \mid C|x, 0\rangle = a_{x,0} |0\rangle \otimes |\Psi_X, 0\rangle + a_{x,1} |1\rangle \otimes |\Psi_X, 1\rangle \text{ with } |a_{x,1}|^2 \geq 2/3\}$
   
   $q\text{AEA}_N = \{C \mid C|x, 0\rangle = a_{x,0} |0\rangle \otimes |\Psi_X, 0\rangle + a_{x,1} |1\rangle \otimes |\Psi_X, 1\rangle \text{ with } |a_{x,1}|^2 \leq 1/3\}$

is a PromiseBQP-complete problem.

**Exercise 3.**

*Logarithmic advice.*

Recall that $P/\log$ refers to the class of languages decided by a polynomial time Turing machine with an advice string of length $O(\log n)$.

1. Write the mathematic definition of $P/\log$.
2. Show that $P/\log$ contains undecidable languages.
3. Show that if $\text{NP} \subseteq P/\log$, then $P = \text{NP}$.

**Exercise 4.**

You saw during the course that $P/\text{poly}$ (and $P/\log$) contains undecidable languages.

*Show that there exists decidable languages outside $P/\text{poly}$.*

*Hint: Use a diagonalization argument over circuits of size $n^{\log n}$.*