1 Question 1

Intuition

Suppose we have a problem $X \in RP \wedge co\text{-}RPP$, then we know that there is a RPP algorithm A and a co-RPP algorithm A' for X. We will show that there is an algorithm B that uses A and A' to solve X in expected polynomial time, with 0 probability of error, thus showing that $RPP \wedge co\text{-}RPP \subseteq ZPP$.

Suppose x is an instance of X. Then, there are two cases. Either x is a YES instance, or a NO instance.

- x is YES: By definition $A(x) = YES$ with probability $\geq \frac{1}{2}$, and NO otherwise. Note that YES responses by RPP algorithms are always correct. Thus, we can keep on running $A(x)$ until $A(x) = YES$, in which case we have the correct answer and we terminate the algorithm.
- x is NO: By definition $A'(x) = NO$ with probability $\geq \frac{1}{2}$, and YES otherwise. Note that NO responses by $co\text{-}RPP$ algorithms are always correct. Thus, we can keep on running $A'(x)$ until $A'(x) = NO$, in which case we have the correct answer and we terminate the algorithm.

In the absence of either algorithm, we will not be able to be fully confident in our answer in one of the cases. For example, if given only A, then in the case where x is NO, $A(x) = NO$ always. But we cannot be sure if x is indeed NO as if x were a YES instance, then NO could be indefinitely outputted anyway.

Algorithm

Run A and A' one after the other. Repeat running the algorithms until $A(x) = YES$ or $A'(x) = NO$, in which case we return YES and NO respectively.

Algorithm 1 ZPP Algorithm

```
procedure B(x, A, A')
    while true do
        if A(x) = YES then
            return YES
        if A'(x) = NO then
            return NO
```

Analysis

It's easy to see that this algorithm is correct. Since we will only ever terminate if $A(x) = YES$ or $A'(x) = NO$, we will always return the correct answer since both A and A' cannot be wrong on YES and NO outputs, respectively. The algorithm will produce the correct output on YES and NO problem instances, or run indefinitely. However, we will show that the expected number of runs in both YES and NO instances is a constant.

Let's calculate the expected number of runs of either algorithm in each case. Let's look at the YES case first. Since we run A and A' repeatedly one after the other, $A(x) = YES$ will be either on the first, third, fifth, etc. run(s). The result is very similar in the NO case. The only difference is that $A'(x) = NO$ could report NO either on the second, fourth, or sixth run(s). The summation for the expected number of runs is as follows (note that the probability of $A(x) = YES$ or $A'(x) = NO$ for YES and NO instances respectively is at least $\frac{1}{2}$).

$$
\begin{aligned}
E(\#runs\ x\ is\ YES) &\leq \sum_{i=0}^{\infty} \frac{2i+1}{2^{i+1}} \\
&\leq \sum_{i=0}^{\infty} \frac{i}{2^i} + \frac{1}{2} \sum_{i=0}^{\infty} \frac{1}{2^i} \\
&\leq 2 + \frac{1}{2} \\
&\leq \frac{5}{2} \\
E(\#runs\ x\ is\ NO) &\leq \sum_{i=0}^{\infty} \frac{2i}{2^i} \\
&\leq 2 \sum_{i=0}^{\infty} \frac{i}{2^i} \\
&\leq 2 \times 2 \\
&\leq 4
\end{aligned}
$$

* Note that $\sum_{i=0}^{\infty} \frac{1}{2^i} = 1$ from the result in class. To determine $S = \sum_{i=0}^{\infty} \frac{i}{2^i}$, let's first consider $\frac{S}{2} = S - \frac{S}{2} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots = \sum_{i=0}^{\infty} \frac{1}{2^i} = 1$. Thus $S = 2$.

Since both A and A' run in poly-time by definition, we know that the expected number of runs in either case is ≤ 4. Thus, the expected run time of B is $4 * O(poly)$, which is still poly-time.

Now, we have an algorithm B that is always correct and runs in expected poly-time, so $B \subseteq ZPP$. Thus, $RPP \land co\text{-}RPP \in ZPP$.