1 Question 2

a)

Let's first prove that $b_j + b_{j+1} > 1$. Since the algorithm is online, suppose that we are at the point where bin j is still open and we have not created bin $j+1$ yet. Suppose that an item s_k comes such that it cannot fit in bin j. This means that bin j should be closed with a total amount of b_j (total sum of items in bin j). Now, the only way such that s_k could not fit into bin j is if $s_k > 1 - b_j$ (more than the remaining space is necessary). We must put s_k into bin $j+1$ instead. Now, we have $b_{j+1} \geq s_k > 1 - b_j$. Adding b_j to this inequality, we have $b_j + b_{j+1} > 1 - b_j + b_j > 1$. Thus, our initial claim is proven.

Now, let's suppose we have m bins after the last item is inserted. Let's find a strict lower bound for $\sum b_i$. Break it into two cases where m is odd or m is even. Suppose m is even. Observe that $b_i + b_{i+1} > 1$ for all $i \in \{1, 3, 5, m-1\}$ by the above proof. By grouping subsequent bins into $\frac{m}{2}$ pairs, where the sum of each pair is more than 1, we establish that $b_1 + b_2 + \ldots + b_{m-1} + b_m > 1 + 1 + \ldots + 1 > \frac{m}{2} > \frac{m-1}{2}$. Suppose that m is odd instead. If this were the case, then we can once again establish that $b_1 + \ldots + b_m > \sum_{i \in \{1,3,m-2\}} b_i + b_{i+1} > \frac{m-1}{2}$ by ignoring the value of last bin b_m and grouping the even number of $m-1$ bins into subsequent pairs, which yields $\frac{m-1}{2}$ pairs, whose sums are more than 1 (each pair). These two cases tell us that $b_1 + b_2 + \ldots + b_m > \frac{m-1}{2}$ no matter the m.

Observe that $\sum b_j \leq OPT$ as the sum of all items/bin values represents an "ideal" packing scenario in which there is no fragmentation, so the number of bins required in this best, idealized case is at least $\lceil \sum b_j \rceil$. Thus, $OPT \geq \lceil \sum b_j \rceil$. In all other cases where there is fragmentation, the optimal solution must still be greater than or equal to $\lceil \sum b_j \rceil$ as this is the theoretical minimum for storing the total value of items. Thus $OPT \geq \lceil \sum b_j \rceil \geq \sum b_j$. (This result was also stated in class.) Since $\frac{m-1}{2} < \sum b_j \leq OPT$, we have $\frac{m-1}{2} < OPT$, and so $m < 2OPT + 1$. Thus, the algorithm uses at most $2OPT + 1$ bins.