a. The greedy algorithm taught in class with weight limit still set to W, but only run the sort by density option instead of both sort by density and sort by value; and instead of excluding the first item j that makes current weight exceed maximum allowed weight W, include that item.

Proof of correctness: including the first item j that makes our weight exceed W does not make our weight exceed $\frac{3}{2}W$ because no item is too big (i.e. $w_i \leq \frac{W}{2}\ \forall\ i$). Before including j, the current weight is $\leq W$, adding $w_j \leq \frac{W}{2}$ makes total weight $\leq \frac{3}{2}W$. This new algorithm also makes our total value $\geq V^*$ because as proved in lecture, including a fraction of item j such that the total weight is equal to W when running by sorted density makes our value equal to the optimal value V^*, including the entirety of j thus makes our total value $\geq V^*$.

Runtime Analysis: We learned that the greedy algorithm is $O(n \log n)$. The only modification to the algorithm is not running the sort by value option, which actually makes the runtime smaller. It's still $\boldsymbol{O(n \log n)}$.

b. Our algorithm:

1) Set weight limit to $\frac{3}{2}W$. Run the greedy algorithm on small items only (i.e. items with weight $w_i \leq \frac{1}{2}W$).
2) Start over. Set weight limit to $\frac{3}{2}W$ again. Put a big item in front, then run the greedy algorithm with small items only; run this for all big items.
3) Return the best run from the above two steps.

Proof of correctness: Observe that the optimal solution when the weight limit is W has 0 or 1 big items, as a big item's weight $w_b > \frac{W}{2}$. If the optimal solution has 0 big items, then step 1 has a satisfying solution because it's part a's algorithm. If the optimal solution has 1 big item, then there exists a satisfying solution from step 2 because we exhaust all options of big items in the beginning. The big item's weight $\frac{W}{2} < w_b \leq W$, so running the greedy algorithm on small items only after putting a big item in the front (step 2) is basically part a but with weight limit $\frac{3}{2}W - w_b$.

Runtime Analysis: step 1 is $O(n \log n)$ because it's the algorithm taught in class. Step 2 is $O(n) * O(n \log n)$ because the number of big items is $O(n)$. Overall, this algorithm is $\boldsymbol{O(n^2 \log n)}$.