CSCE 411: Mastery Quiz 6

Name: _____

UIN: _____

You want to mail a collection of n collectible figurines. You will pack them into boxes, and your goal is to use as few total boxes as possible while satisfying the following constraints:

- Each box has room for at most two figurines.
- Each figurine has a weight, and the total weight of each box must be at most k.

Give an efficient algorithm that takes as input an integer k representing the weight limit and an array $A[1, \ldots, n]$ of figurine weights and outputs the minimum number of boxes that you need to ship these figurines. You may assume that $\max_{i=1}^{n} A[i] \leq k \leq 2 \max_{i=1}^{n} A[i]$ (i.e. that the heaviest figure weights at most k and that k is at most double the heaviest figure's weight).

For example, given $A = [4, 12, 2, 19, 5, 6, 7, 18, 17]$ and $k = 20$, you should output 6. A packing which achieves this minimum number of boxes is $[4, 12], [2, 18], [19], [5, 6], [7], [17]$

You must prove the correctness of your algorithm. Analyze the runtime of your algorithm.

Solution: We give the following algorithm:

```
ShipFigures(A, k)
  Sort A in decreasing order
  While there are figures left to pack:
      If the heaviest and lightest figure can be packed together:
          Pack them together
      Else:
          Pack the heaviest figure by itself
  Return the number of boxes used
```

Correctness: We need to prove that our algorithm uses the minimum possible number of boxes. Let ALG be the collection of packed boxes used by our algorithm. We will prove that ALG is optimal by induction on n.

- Base Case: If $n = 1$, there is only one possible packing, put the figure in its own box.
- Induction Hypothesis: Suppose that for any list of fewer than n figures and any weight limit k, ShipFigures uses the minimum possible number of boxes.
- Induction Step: Suppose $n > 1$. Let f be the heaviest figure and g be the lightest figure. If they fit in the same box, ALG will pack them together, otherwise it will pack f by itself. We consider both cases:
  - If ALG packs f by itself, it means that no other figure fits with f. Therefore, any packing of these figures must pack f by itself. Let A' be the list of figures with f removed. Let ALG' be the packing that ShipFigures uses for A'.By our IH,ShipFigures finds the optimal number of boxes for A'. Since every packing for A must put f by itself, and $|ALG| = |ALG'| + 1$, this means that ALG uses an optimal number of boxes for A.
  - If ALG packs f and g together. We claim that there is an optimal packing for A that also packs f and g together. Let OPT be an optimal packing and suppose that OPT does not pack f and g together. If OPT packs f by itself, we can move g to f's box to get a new packing with the same number of boxes which packs f and g together. If OPT packs f with some other figure h, then we can swap g and h. Since f is the heaviest figure, the figure h is paired with after the swap cannot be heavier than f, and since g is the lightest figure, f's box is not heavier after the swap either. Thus, this new packing is valid. Thus, we can assume OPT packs f with g.

    This means that an optimal packing for A uses 1 more box than an optimal packing for $A' := A$ with f and g removed. By our IH, ShipFigures finds an optimal packing for A', and ALG consists of the $f - g$ box plus the packing the algorithm uses for A', so ALG is an optimal packing for A.

Runtime: Now we analyze the runtime of our algorithm. Sorting A can be done in $O(n \log n)$ time. The while loop runs for $O(n)$ iterations, and each iteration can be performed in constant time (since the array is sorted, we can simply use a pair of indices to keep track of the heaviest and lightest figure we have yet to pack). Thus the total time for the while loop is $O(n)$ and so the total runtime for the algorithm is $O(n \log n)$.