## Step 1. Get initial solution by greedy algorithm.

This step uses a simple greedy algorithm to get approximate solution which is used in backtracking search.

* Sort numbers in descending order and add numbers to partitions one by one.
* Every number is inserted to the smallest partition i.e., the partition with least sum
* After all numbers are inserted to partition, difference with largest and smallest sums is calculated and stored as candidate solution.(bestDiff)

## Step 2. Backtracking with A\* cutting algorithm

Recursive search function findSolution uses A\* algorithm to reduce running time.

*A\* algorithm – check current state and candidate solution. When it can be determined whether current state is definitely worse than candidate solution, recursive function does not search anymore and exits immediately.*

Function “check” is A\* check function that check current min & max sums and candidate. Return value TRUE of this function means that you don’t need search current state anymore and hence findSolution function returns immediately.

If return value is FALSE, it means better solution *may be* found and findSolution searches all possibilities.

## Step 3. Build solution

Same as prev solution