Question 1

There is an old desalination plant close to your residential area. It filters out the salt from the ocean in a filtration tank. The tank takes in water from the ocean through pipes but only accepts intake pipes of a given size k.

You are given a list S of n pipes, with varying capacities less than k, that pump water from the ocean. You could simply feed each pipe in S into it's own intake pipe at the filtration tank, but this would require n intake pipes. You figure you can reduce the number of intake pipes needed at the filtration tank by feeding two smaller pipes, whose combined capacity is $\leq k$, into a larger pipe which is then attached to a single intake pipe.

Your task as the on-site engineer is to find the smallest number of intake pipes needed to connect all the pipes in S to your filtration tank.

For example, The list of pipes S:

$$\begin{bmatrix}1, & 5, & 2, & 8, & 3, & 6, & 3, & 6, & 3\end{bmatrix}$$

If we choose a maximum intake capacity of $k = 10$, then the minimum number of intake pipes after combining is 4.

We are able to combine the pipes as follows:

$$\begin{bmatrix}1, & 5, & 3\end{bmatrix}, \begin{bmatrix}2, & 8\end{bmatrix}, \begin{bmatrix}6, & 3\end{bmatrix}, \begin{bmatrix}6, & 3\end{bmatrix}$$

1.1 One of your fellow engineers, James, proposes a greedy algorithm that would solve this: Take the largest pipe that is smaller than k and combine it with the next largest pipe such that their sum is still $\leq$ k. Repeat until you cannot combine any more pipes. Now you have combined the pipes optimally so the minimum number of intake pipes required is the total number of pipes left at the end.

This algorithm seems correct but you are not certain. It solves the example above by grouping the pipes as: $[8, 2], [6, 3], [6, 3], [5, 3, 1]$ which gives the correct result of 4. Formally prove whether or not this algorithm always achieves the correct result.

Unfortunately James' algorithm is not always correct. We offer a proof by counterexample that this algorithm will sometimes make a sub-optimal choice of pipes to combine.

For $S = [2, 2, 2, 3, 5, 6]$ and $k = 10$, James' algorithm would combine the pipes to get $[6, 3], [5, 2, 2], [2]$ giving a result of 3 intake pipes. However, the correct result is actually 2 as we may join the pipes the following way $[6, 2, 2], [5, 3, 2]$.

From this we can clearly see that James' algorithm does not achieve the optimal result in all cases and is therefore not correct.

- **[1 mark]** Identifies that algorithm is not always correct
- **[1 mark]** Correct proof as to why the algorithm is incorrect

1.2 You realise that the correct solution is very slow and the pipe management would be a disaster. Instead, you decide that you will only combine pipes if they are adjacent to each other in S.

For example, if our list of pipes was $S = [1, 5, 1, 2, 8, 3, 6, 3, 6, 3]$, and our maximum intake capacity was $k = 10.$, then the minimum number of intake pipes is 5.

We may combine adjacent pipes as follows: $[1, 5, 1], [2, 8], [3, 6], [3, 6], [3]$.

Describe a greedy algorithm that computes the minimum number of intake pipes needed with this new constraint and give its time complexity.

We may simply start from the leftmost pipe and combine it with as many possible pipes to the right until doing so would increase the capacity above k. We then start with the first pipe that was not combined and repeat the process until we have moved through the entire list. The minimum number of intake pipes is the number of elements in the final list. This algorithm moves through the whole list once and so may be done in $O(n)$.

- **[2 mark]** Gives a greedy algorithm that arrives at the correct solution.
- **[1 mark]** Provides time complexity and some justification.

1.3 Formally prove that your algorithm from **1.2** achieves the correct result.

Proof: (Using exchange arguments) We will consider the greedy algorithm as described above.

Let $A = \{i_1, ...i_x\}$ be the set of resulting pipes (each of which may be a combination of smaller pipes that sum to it) produced by the greedy algorithm. Let $O = \{j_1, ...j_x\}$ be the optimal solution, which differs from the greedy algorithm. Thus, $|O| <= |A|$.

Since O differs from A, at some point, the optimal solution did not combine the current leftmost pipe with the pipe to its right, even though their sum would've been $\leq k$.

Let k be the first instance where O differs from A, then we have:

A: $\{i_1,\, i_2,\, ..i_{k-1},\, i_k,\, i_{k+1},\, ...i_x\}$
O: $\{i_1,\, i_2,\, ..i_{k-1},\, j_k,\, j_{k+1},\, ...j_y\}$

Since A always combines the elements to the immediate right of the pipe in consideration until their sum is $\leq k$; $|i_k| >= |j_k|$ since the greedy solution aims to maximize the number of pipes in each combined pipe.

We can then use the exchange argument to show that replacing j_k with i_k by combining the first elements of j_{k+1} with j_k instead will result in no loss in total number of intake pipes. Thus:

A': $\{i_1,\, i_2,\, ..i_{k-1},\, i_k,\, i_{k+1},\, ...i_x\}$
O': $\{i_1,\, i_2,\, ..i_{k-1},\, i_k,\, j'_{k+1},...j_y\}$

Therefore, by making this exchange for every such instance, the resulting set O' would have $|O'| \leq |O|$. Eventually $O' = A$ and hence A must also be an optimal solution and produce the set with the smallest number of intake pipes needed.

- **[2 mark]** Explains the proof correctly, arrives to the conclusion that the greedy solution is optimal.
- Any other proof techniques with a valid result allowed (for ex. greedy always ahead).