1. Represent the \textsc{Insert}-operation for B$^+$-tree as pseudocode. Use the method of splitting all full nodes as you go down in the tree (as opposed to starting the splitting from the leaves). As a model you can use the algorithm in the Cormen et al. book, but notice that they use “regular” B-trees.

2. You are given a sequence of positive integers $a_1, \ldots, a_n$ as input. The task is to decide, whether the sequence can be split into two parts in such a manner that each number will be in exactly one of the parts and the sum of numbers in each part is the same. In other words, you must decide whether there is a set $A \subseteq \{1, \ldots, n\}$ such that \[
\sum_{i \in A} a_i = \sum_{i \notin A} a_i.
\]

Examples:
- Input $(a_1, a_2, a_3, a_4, a_5) = (21, 25, 13, 12, 39)$: the answer is “yes”, because $a_1 + a_5 = 21 + 39 = 25 + 13 + 12 = a_2 + a_3 + a_4$ (so we could choose e.g. $A = \{1, 5\}$).
- Input $(a_1, a_2, a_3, a_4, a_5, a_6) = (15, 7, 15, 5, 3, 15)$: the answer is “yes”, because $a_1 + a_3 = 15 + 15 = 7 + 5 + 3 + 15 = a_2 + a_4 + a_5 + a_6$ (so we could choose e.g. $A = \{1, 3\}$).
- Input $(a_1, a_2, a_3, a_4, a_5, a_6) = (11, 17, 47, 5, 4, 33)$: the answer is “no”, because no matter how you split the numbers, the sum of one part will be even and the sum of the other part odd.

Give a backtracking algorithm for the problem. How do you represent partial solutions, and what are the options for further developing a given partial solution? How do you recognise non-promising solutions? Estimate roughly for the time and space complexity of your algorithm.

Remark: This problem is known with the name \textsc{Partition}. It is NP-complete, i.e. in the same complexity class with the Travelling Salesman Problem and similar problems.

3. Keys 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 are inserted into an initially empty B$^+$-tree (in that order). Show the end result and main intermediate steps, when the maximum size of a node is $2t = 4$.

4. Consider the B$^+$-tree in the picture below.

```
            28
          /   \ 
       132   22
      /   \     \ 
     7 12 13 15 18
    / \   / \   / \\
   24 25 27 33 36 37
```

Show the resulting trees, when we first remove the keys 57, 12, 41, 27, 25 and 24. (The keys are to be removed sequentially in that order: use the result of the previous operation as the starting point for the next one.)