1. The partition problem consists of determining if there is a way to partition a set of integers \( S \) into two subsets \( S_1 \) and \( S_2 \) such that \( \sum S_1 = \sum S_2 \). Recall that \( S_1 \) and \( S_2 \) are a partition of \( S \) if and only if \( S_1 \cup S_2 = S \) and \( S_1 \cap S_2 = \{\} \). Write a method that solves the partition problem using backtracking.

2. Write a randomized algorithm to check probabilistically whether two trigonometric expressions are identities. Your expression can be written as java methods. For example, to check whether \( \tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)} \) you can define methods:

```java
public static double f(double theta){
    return Math.sin(theta)/Math.cos(theta);
}

public static double g(double theta){
    return Math.tan(theta);
}
```

And then implement a method that tests the equality using a large number of random values in the \(-\pi\) to \(\pi\) range. Try your method on as many of the identities in the following image as you can. Try it also on expressions that are not identities to make sure those are correctly detected.
As usual, write a report describing your work.