Problem 4: Avoid circular routes in the forest 10%

Start with the classes you wrote in problem 1 (make a copy). You should now avoid circular routes in the structure of paths and crossings that you create; in other words, it should now be impossible for a walker to come to a crossing previously visited. This is achieved by using a recursive method which checks whether a route already exists between the two crossings selected when you create a new path.

Write a recursive method in the class Kryss: boolean finnesVeiTil(Kryss kryss2) ("is there a path to"). As you did previously, the constructor in Skog shall, for each new path created, select two random crossings: kryss1 and kryss 2. Then, you shall call finnesVeiTil() in one crossing with the other crossing as parameter. If a route of paths already exists from kryss1 to kryss2, you shall not save this path (as that would have produces a circular route); instead, you shall select two other crossings and try those. If you have not found two satisfactory crossings after 100 tries, you are to abandon the search and terminate with an error message.

The two crossings at opposite ends of a path are called *neighbour crossings*. If kryss1 calls the Boolean method finnesVeiTil(kryss2) ("isThereAPathTo") in itself (this), the method shall return true if kryss2 is the same crossing as kryss1, or if any of the neighbour crossings of kryss1 has a route to kryss2. Otherwise, it returns false.

Remember to ensure that the crossing does not ask the neighbour from which the call came, as the recursion would never end. You might want to add a parameter to the method finneVeiTil() to avoid this.

When simulating such a forest with circular routes, a walker may still decide to exit a crossing using the path on which he or she arrived.

Problem 4 weight 10%

Write the method finnesVeiTil() in the class Kryss.
Make the described changes to the constructor of Skog
End problem 4

Uploading problem 4: Upload a zip file containing the modified classes Skog og Kryss. If you have modified other classes to solve problem 4, include these in the zip file as well, and add a comment indicating what has been altered. The name of the zip file should be Oppgave4.zip.