Programming Assignment #1
Brute-Force Set Cover

CS224 Fall 2022
15 points
due Tuesday, Sept. 13th, 11:59 pm

1 Set Cover

Solve a set-covering problem by brute force.

1.1 Instructors and courses

In the art department of some particular college, there are twelve instructors and ten separate courses. Each instructor can teach one or more of the courses. This table shows the twelve instructors and which courses each is capable of teaching:

Instructor	ART001	ART002	ART008	ART064	ART110	ART124	ART125	ART201	ART205	ART266
John		✓	✓				✓			
Tom		✓			✓		✓			
Mary			✓							✓
Alicia				✓	✓		✓			✓
Betsy						✓	✓	✓		
Kira			✓							✓
Hiram	✓				✓		✓			
Simon	✓		✓				✓			
Viggo					✓	✓	✓			
Ralph			✓	✓					✓	✓

Find the minimum covering set: the smallest set of instructors such that for every course c, the set contains at least one instructor capable of teaching c. Do this by a brute-force technique: try every possible subset of instructors.

I've created infrastructure for you, in the `BruteForce` folder of the course gitlab: `Main.java` and `Instructor.java`. Use these.

Write this function: `ArrayList<boolean[]> permute(int n)`. This will return 2^n arrays, each of length n, with all possible combinations of `true` and `false`. For example, `permute(3)` will return an `ArrayList` containing these eight arrays: `[false, false, false]`, `[false, false, true]`, `[false, true, false]`, `[false, true, true]`, `[true, false, false]`, `[true, false, true]`, `[true, true, false]`, `[true, true, true]`.

Here's the algorithm:

```
// initialize a new ArrayList<boolean[]> rtnval
// this will be the return value of the function
if (n == 0) {
  // create a new boolean[] array of length = 0
  // add this array to rtnval
} else {
```

```
  sublist = permute(n-1)
  for each element e of sublist {
    create a new boolean[] array a1 of length n
    copy e to a1
    set a1[n-1] to true
    add a1 to rtnval
    create a new boolean[] array a2 of length n
    copy e to a2
    set a2[n-1] to false
    add a2 to rtnval
  }
  return rtnval
}
```

Write this function in `Main`:

```
Instructor[] findMinCover(String[] courses, Instructor[] instructors)
```

It should return an array consisting of the smallest number of instructors necessary so that each course has an instructor who can teach it.

The function `testOne()` in `Main.java` shows how to set up a schedule and call `findMinCover()`.

Write a function `testTwo()` on `Main` that sets up the schedule represented by the table above and then calls `findMinCover()` to find minimum-size covering set.

2 What to Submit

Submit your `Main.java`.

3 Graduate Students

Graduate students, and undergraduates who would like a little extra credit: sort the `ArrayList` that `permute()` returns, in ascending order by the number of `true` values in each array. Then, you can stop checking after the first covering set you find.