Practicum 3
4/10/2023

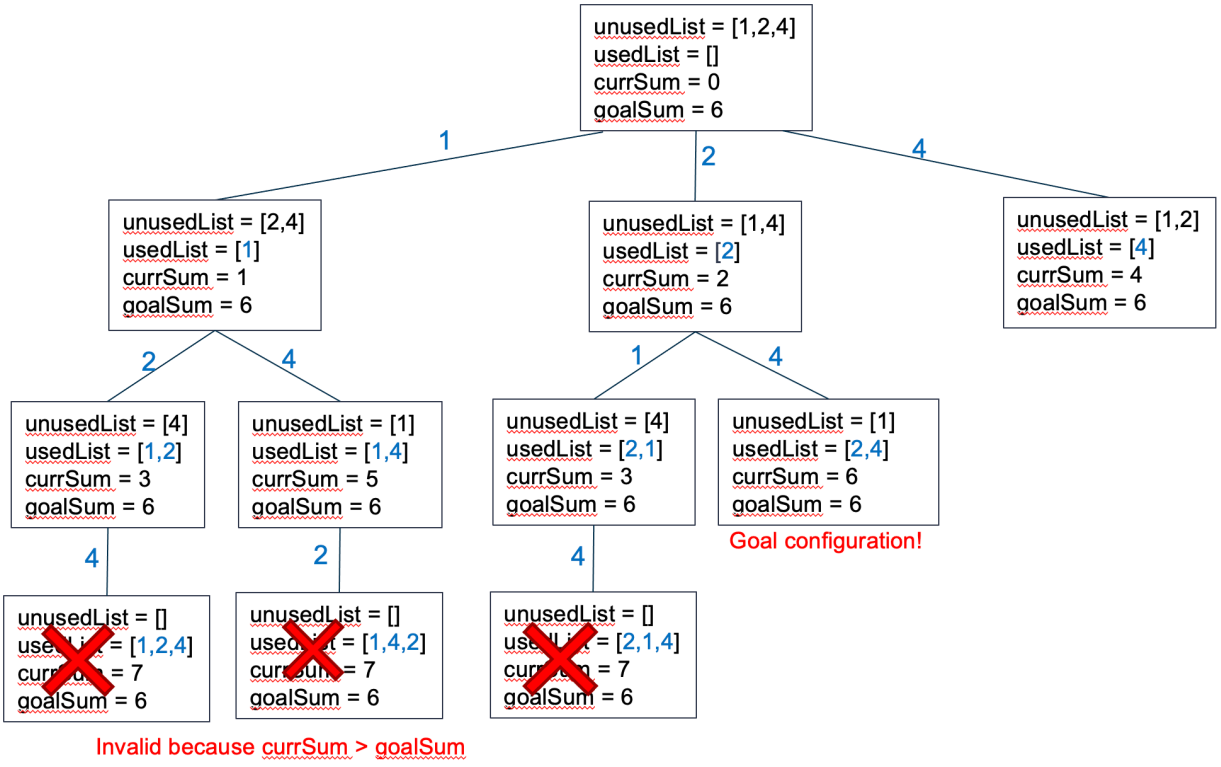
Note: The total score is 110 points (= 50 + 35 + 25). However, the maximum grade you can receive will be capped at 100 points.

Problem #1: Backtracking (50 points)

Given a list of integers and sum, the objective is to find a combination of integers that adds up that sum. Examine the provided class **`Problem1.java`** and complete the following methods:

- `getSuccessors()`
- `isValid()`
- `isGoal()`

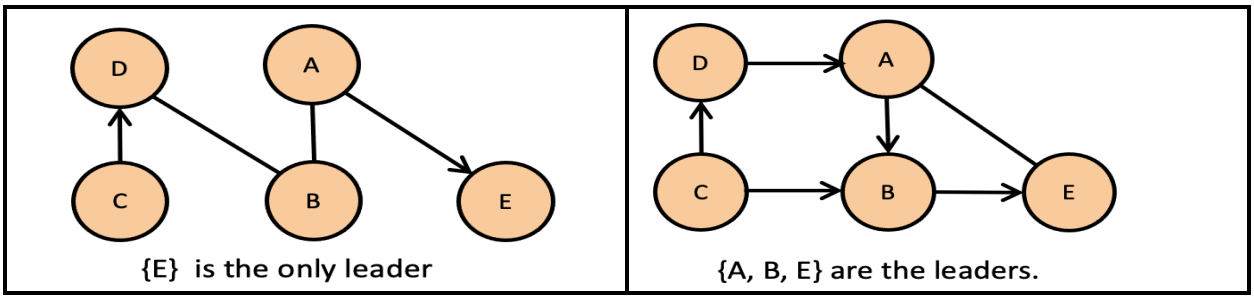
Example:
Given a list `[1,2,4]` and `sum = 6`, find a combination of integers that adds up 6.



Problem #2: Graph Search (35 points)

A ***leader*** is a vertex in a graph that can be reached from ***all*** other vertices by at least one path. There may be more than one leader vertex in a graph! In the provided `problem1.java` file, complete the `findLeaders()` method using the `graphs.AdjacencyGraph` class. In

Problem2. `main()`, create the graph depicted in the **second example** below to test your `findLeaders()`.



Hint: Use either the `bfSearch()` or `dfSearch()` in the `.graphs.AdjacencyGraph` class.

Problem #3: Anonymous classes and Lambdas (25 points)

1) Examine the provided file **Problem3**.java and the following class:

```
public class Example implements Testable {
    @Override
    public boolean test(int elt) {
        return elt < 10 && elt > -10;
    }
} // not in the provided file
```

If you pass `list = [7, -15, 9, -2, 8, 0, -3, 20, -12, 5]` and an `Example` object into the `myfilter` function, i.e., `myfilter(list, new Example());` what would be returned? Include your answer as a comment in the `main` method.

2) In `main()`, use an anonymous class to initialize a `Testable` object named `isPositive`:

```
Testable isPositive = <an anonymous class>;
```

The test function in the anonymous class must return `true` if the specified value is a positive number, and `false` otherwise.

3) In `main()`, use a lambda expression to initialize a `Testable` object named `isEven`:

```
Testable isEven = <a lambda expression>;
```

The lambda must return `true` if the specified value is even, and `false` otherwise.