

- Variante 1

Lösung Teil 1

Die heutige Aufgabe war ziemlich schwer zu verstehen und dadurch auch noch schwerer zu erklären. Auch mangels Zeit gibt es daher für diese Variante 1 leider nur einen Lösungsvorschlag. Diese Lösung ist noch dazu etwas verzwickelt nachzuvollziehen, da sie als Haupt-Datenstruktur eine ArrayList von ArrayLists von long-Arrays nutzt (ArrayList<ArrayList<long[]>>). Dafür ist sie aber sehr effizient und verzichtet quasi auf jegliche Schleifen (von der direkten Länge der Eingabedaten abgesehen).

Lösungsvorschlag

```
private long[] numbersToIntArray(String numbersStr) {
    String[] numbers = numbersStr.trim().split(" ");
    long[] result = new long[numbers.length];

    for (int i = 0; i < numbers.length; i++) {
        result[i] = Long.parseLong(numbers[i]);
    }

    return result;
}

private long getMapping(ArrayList<long[]> map, long input) {
    for (long[] line : map) {
        if (input >= line[1] && input < line[1] + line[2]) {
            return line[0] + (input - line[1]);
        }
    }
    return input;
}

public long partOne() {

    long[] seeds = numbersToIntArray(inputLines.get(0).split(":")[1]);
    ArrayList<ArrayList<long[]>> allMaps = new
ArrayList<ArrayList<long[]>>();
    ArrayList<long[]> map = new ArrayList<long[]>();

    for (int l = 2; l < inputLines.size(); l++) {
        String line = inputLines.get(l);

        if (line.length() <= 1) {
            allMaps.add(map);
            continue;
        }

        if (line.contains(":")) {
            map = new ArrayList<long[]>();
            continue;
        }
    }
}
```

```
    }

    map.add(numbersToIntArray(line));

}
// add last map to allMaps
allMaps.add(map);

ArrayList<Long> seedResults = new ArrayList<Long>();
for (long seed : seeds) {
    for (int mapCount = 0; mapCount < allMaps.size(); mapCount++) {
        seed = getMapping(allMaps.get(mapCount), seed);
    }
    seedResults.add(seed);
}

long smallest = seedResults.get(0);
for (int i = 1; i < seedResults.size(); i++) {
    if (seedResults.get(i) < smallest) {
        smallest = seedResults.get(i);
    }
}

return smallest;
}
```

From:
<https://www.info-bw.de/> -

Permanent link:
<https://www.info-bw.de/faecher:informatik:oberstufe:java:aoc:aco2023:day5:start?rev=1701788397>

Last update: **05.12.2023 14:59**

