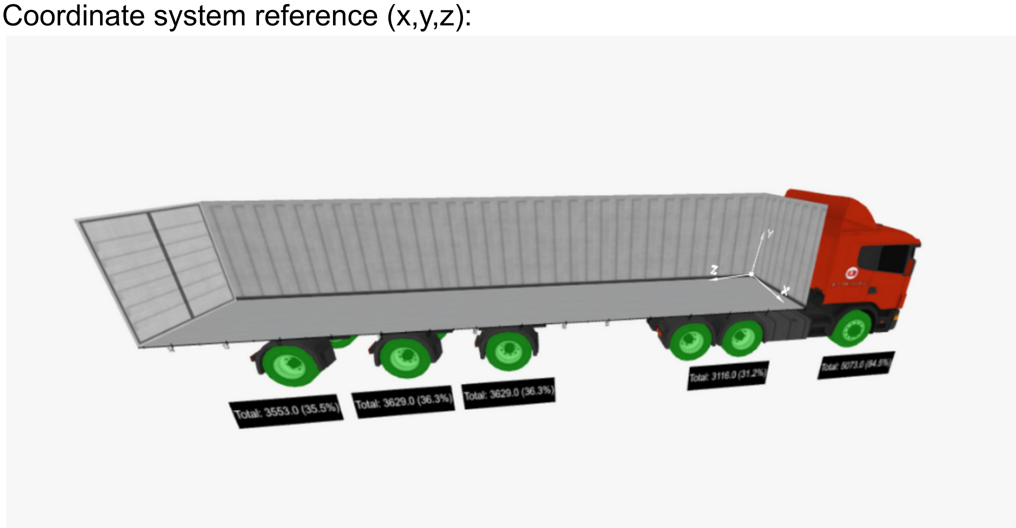
Instructions

- The solution must be sent within **<u>1</u>** week starting when the email with this document is sent;
- Do your best, we will be happy to review your solutions even if it is incomplete;
- Beyond this document you'll be receiving **<u>2</u>** input files to use;
- Use GIT!
- Preferably use one of the following programming languages:
  - C++, Java, Python, R;
  - <u>Justify if you use other</u>;
- The following points will be evaluated:
  - Natural language communication skills;
  - Objectivity;
  - Modeling skills:
    - Modeling tools;
    - Modeling techniques;
    - Mathematical language;
    - Modeling standards;
  - Programming skills:
    - Code legibility;
    - Code organization;
    - Code standards;
  - Complexity:
    - Architecture complexity;
    - Asymptotic complexity;
    - Solution simplicity;
  - Attention to details;
- Execute each **task** on separated file/folder on a GIT repository;
- **We are open to help you with your questions via email.**
- Coordinate system reference (x,y,z):



Merge Problem

Union of the allocatable spaces to find bigger allocatable spaces

Goal

- **Maximize**(**Average**(merged containers volume))

Subject to

1. The union of the input and the union of the output must be the same set (of R^3 points);
2. Container merge can only be done if all parent-containers have equal **y_start**;
3. The containers must be rectangular parallelepipeds/cuboid:
   - They have lateral edges perpendicular to the base, that is, they have right angles (90º) between each of the faces;
4. Containers can't be inside or have equal coordinates of others containers in the output:
   - There must not exist two containers C_1, C_2 such that $C_1 \subseteq C_2$;
5. The output is limited to 3x of the input size:
   - When input has 200 containers, the output is limited to 600 containers;

Contract

Allocatable space are what we call "container", Input and output are Lists of "containers";

Each container C can be seen as an R^3 set defined:

- $C = [x_{start}, x_{end}] \times [y_{start}, y_{end}] \times [z_{start}, z_{end}] \subseteq \mathbb{R}^3$, being $\times$ the cartesian product.
- $volume_C = |x_{start} - x_{end}| \cdot |y_{start} - y_{end}| \cdot |z_{start} - z_{end}| \subseteq \mathbb{R}$

Input

- **JSON Array:** Allocatable 3d space, each with three coordinates from where it begins the space and three coordinates from where it ends the space and finally, each will have an identifier.

Output

- **JSON Array:** Merged containers, each with three coordinates from where it begins the space and three coordinates from where it ends the space and finally, each will have an identifier.

Input/Output Example

```
[
    {
        "id": 0,
        "x": {"start":0, "end":2.38},
        "y": {"start":0, "end":2.83},
        "z": {"start":0, "end":12.0},
    },
    ...
]
```

Tasks

1. Describe clearly with natural language step by step your(s) chosen solution(s);

2. Use diagrams and/or machine states to document your modeling phase;

3. Code your solution, It will be expected the code of solution and 2 outputs each for a correspondent input:
    a. case_1_input.json → case_1_output.json
    b. case_2_input.json → case_2_output.json

4. Calculate the *asymptotic complexity* of your implementation;