

## Første porteføljeopgave: Palletering

Du er ansat i en virksomhed, som laver og sælger en automatiseret palleteringsløsning. Jeres firma bruger en kollaborativ robot og en suction-gripper til at gribe papkasser og stable dem på europaller. Jeres system henter automatisk en liste af papkasser, der skal palleteres, fra kundens Enterprise Resource Planning (ERP) system, som en JavaScript Object Notation (JSON) fil. Filen består af en liste af kasser, hvor hver kasse er repræsenteret som et JSON objekt: et sæt af key-value par. Hvert objekt har information om kassens id, order id, størrelsen på kassen (højde, bredde, dybde) i meter, vægten i kilogram, og et destinations id, der har en tilhørende adresse i kundens ERP system.

Bemærk at sættet af key-value par her betegnes som et objekt, fordi det nævnes i en JSON kontekst, men kunne også betegnes som en Dictionary (python) eller et Map (c++, Java).

Herunder et eksempel på en JSON fil med tre kasser:

*boxes.json*

```
[
  {"box_id": 15, "order_id": 299, "size": [0.5, 0.1, 0.3],
   "weight": 04.2, "dest_id": 19},
  {"box_id": 34, "order_id": 313, "size": [0.5, 0.2, 0.4],
   "weight": 01.2, "dest_id": 19},
  {"box_id": 103, "order_id": 299, "size": [0.2, 0.9, 0.5],
   "weight": 10.1, "dest_id": 19},
]
```

Indtil videre har jeres system stablet kasserne i den rækkefølge, der er i filen, og det har ikke tidligere været noget problem, fordi kunden kun har sendt kasser af den samme størrelse afsted til den samme kunde på samme ordrenummer, men nu har kundens behov ændret sig.

Kunden vil gerne have, at alle kasser på en palle har **den samme destination** og tilsammen **vejer maks et ton**. Hvis det er muligt, vil de også gerne samle kasser med **samme ordrenummer på den samme palle**, men ikke hvis det betyder, at der skal sendes flere paller afsted, end hvis de ikke var separeret mht. ordrenummer.

En anden udvikler i dit firma har lavet en algoritme til at stable et sæt af kasser på en palle, hvor den samlede volumen af kasser på en palle maks må være én kubikmeter.

## Opgave

Lav et program, der deler et sæt af kasser ud på flere paller, som respekterer kravene ovenfor. Programmet skal implementeres i Python ved brug af test-drevet udvikling med PyTest og versionsstyring med Git. Lad både produktionskode, testkode og testlisten være en del af versionsstyringen. Lav et commit hver gang i får en ny test til at lykkes (det er ikke en generel opfordring, men bare til opgaven her for at få lavet nogle commits). Det gør ikke noget, at historikken bliver lidt rodet - lad være med at ændre i historikken. I må gerne diskutere opgaven sammen, men i skal implementere al koden selv. Aflever dit repository (".git" mappen) som en zip fil.

Det er ikke forventet, at i bruger avancerede optimeringsalgoritmer.

## Hints

I jeres tests kan det være en fordel at udelade selve JSON filerne og beskrive dem som lister af dictionaries i stedet.

Her er nogle ressourcer, som måske er en hjælp:

- Python datastrukturer (måske især dictionaries)
- Sortering med en key funktion