ASSignment 4

24.09.24

Task 1

Creating an algorithm with complexity O(n) would be done by just applying the given formula in a function.

to achieve the logarithmic term we make the function recursively calculate the center of mass.

Pseudocode:

```
def CG(p: list):
    if particles == 1:
        weight, pos ← p
        return weight, pos

    mid ← len(p)//2
    L ← p[:mid]
    R ← p[mid:]

    mass1, pos1 ← CG(L)
    mass2, pos2 ← CG(R)

    combined_mass ← (mass1 + mass2)
    combined_pos ← $\frac{\sum_{n=1}^{n} w_i \vec{P_i}}{combined\_mass}$

    return combined_mass, combined_pos
```

Task 2

The divide-and-conquer algo is based on a recursive step, a combine step and a counting step.

* Recursive step:

- Split array into halves and recursively detesmine the majority element for each half.

* Combine step:

- if L and R has return same element → that element is majority.
- If they return different element, count the occurences of both elements in the current subarray and return the most frequent one.

* counting step:

- L and R candidate we count amount in combined. array.
  → Element that occurs the most is majority element

⇒ Running time becomes $O(n \cdot \log(n))$
⇒ recursive step is $O(\log(n))$
and counting step is $O(n)$