

به نام خدا

علی غفاری ثابت - ۹۸۲۳۹۰۳

توضیح کد:

۱- تابع group لیستی از جرم قطعات و تعداد کامیون ها را دریافت می کند و اجسام را برای کامیون ها گروه بندی می کند. گروه بندی به این صورت انجام می شود که ابتدا اجسام را بر حسب جرم به صورت نزولی مرتب می کند سپس میانگین این که در هر کامیون چقدر بار باید قرار بگیرد را حساب می کند. در نهایت در هر کامیون هر بار جسم با جرم بیشتر را قرار می دهیم تا مجموع وزن بار کامیون بیشتر از میانگین بدست آمده شود. سپس حاصل را چاپ می کنیم.

۲- در بدنه اصلی کد تعداد کامیون و جرم اجسام را از کاربر دریافت می کنیم و به عنوان ورودی به تابع gorup می دهیم.

پیچیدگی زمانی:

در هر مرحله در تابع group هر جسم در یک کامیون قرار می گیرد و چون n جسم داریم پس پیچیدگی زمانی این الگوریتم برابر $\theta(n)$ است.

اثبات نرخ تقریب:

فرض می کنیم A برابر میانگین جرم اجسام و M جرم سنگین ترین جسم باشد. بنابراین در حالت ایده آل (نه جواب صحیح) حداقل وزن سنگین ترین کامیون برابر $\max(M, A)$ است. وزن سنگین ترین کامیون در حالت ایده آل را W می نامیم. پس وزن سنگین ترین کامیون در حالت صحیح بزرگتر مساوی W است. حال برای هر جسم و حالت داریم: ۱- یا از A بیشتر است پس در یک کامیون جدا قرار می گیرد و سراغ کامیون و جسم بعدی می رویم زیرا جسم قبلی چون سنگین تر بوده در کامیون قبلی قرار گرفته و کامیون فعلی خالی است. در این حالت وزن کامیون قطعا کوچکتر مساوی W است. ۲- وزن جسم فعلی کمتر از A است. پس با قرار دادن آن در کامیون فعلی یا کامیون پر می شود یا همچنان وزن آن کمتر از A می ماند. اگر پر شود که وزن آن کامیون کمتر از $2A$ (زیرا وزن آن جسم کمتر و وزن فعلی کامیون کمتر از A است.) و در نتیجه کمتر از $2W$ است. پس در هر حالت وزن کامیون کمتر از $2W$ است در نتیجه نرخ تقریب در این الگوریتم برابر ۲ است.

تصوير محيط اجرا:

```
ali@Ali:~/Desktop/University/3992/algorithm_design/projects/project4/individual-project-4-aghs8055/post_office$ python3 post_office.py
Enter count of trucks: 4
Enter parts weight: 6 1 3 4 2 5 1 6
Trick #1: 6 6
Trick #2: 5 4
Trick #3: 3 2 1 1
Trick #4:
ali@Ali:~/Desktop/University/3992/algorithm_design/projects/project4/individual-project-4-aghs8055/post_office$ python3 post_office.py
Enter count of trucks: 3
Enter parts weight: 1 2 3 6
Trick #1: 6
Trick #2: 3 2
Trick #3: 1
ali@Ali:~/Desktop/University/3992/algorithm_design/projects/project4/individual-project-4-aghs8055/post_office$ python3 post_office.py
Enter count of trucks: 10
Enter parts weight: 1 2 0 0 0 0
Trick #1: 2
Trick #2: 1
Trick #3: 0 0 0 0
Trick #4:
Trick #5:
Trick #6:
Trick #7:
Trick #8:
Trick #9:
Trick #10:
ali@Ali:~/Desktop/University/3992/algorithm_design/projects/project4/individual-project-4-aghs8055/post_office$ python3 post_office.py
Enter count of trucks: 4
Enter parts weight: 5 2 3 6 9 8 7 5 1 4 5
Trick #1: 9 8
Trick #2: 7 6 5
Trick #3: 5 5 4
Trick #4: 3 2 1
```