تیم AZURE (به معنای لاجوردی) **مدیر تیم: دکتر سیّد محمدرضا لاجوردی**

توضیح شرایط مسئله

n شیء و m جعبه داریم که هر جعبه اندازه‌اش برابر k است. اشیاء به ترتیب از چپ به راست از ۱ تا n شماره‌گذاری شده‌اند و اندازه‌ی شیء iام برابر با a_i است $(\forall i\,;\, k \geq a_i \geq 1)$.

اگر در مسئله برای هر یک از اشیا جایگاه ۱ تا n را به ترتیب j بنامیم، از آنجا که در برنامه این اشیا را در یک آرایه به نام *size_of_object* قرار داده‌ایم و ایندکس آرایه همان معنای j را دارد بنابراین به جای حرف j کلمه index را انتخاب کرده‌ایم.

از آنجا که مسئله یک مسئله ایستا است می‌توان برای یافتن جواب، از طریق مساوی قرار دادن n=index، آخرین جعبه را اول پر کرد و سپس با کم کردن index به جعبه‌ی ماقبل رسید. این کار را تا زمان پر شدنِ در حد امکانِ جعبه‌ها و یا تمام شدن اشیا قبل از جعبه اول ادامه می‌دهیم. اگر مسئله از نوع پویا بود و ما تعداد جعبه‌ها یا اشیا را در میان مسئله تغییر می‌دادیم نمی‌توانستیم از این راه حل استفاده کنیم بلکه باید از اولین جعبه پیش می‌رفتیم. علت حرکت از آخر به اولین جعبه، پیمایشِ کمترین محاسبات برای رسیدن به ماکزیمم اشیای درون جعبه‌ها می‌باشد زیرا در این روش پیچیدگی محاسباتی از نوع O(n) است ولی در روش اول به آخر پیچیدگی محاسباتی برابر n است.

شرط مسئله مبنی بر آنکه حتما تمام شئی‌های j تا n را در جعبه‌ای قرار دهید این مسئله را با مسئله‌ای که به دنبال یافتن ماکزیمم تعداد اشیا درون جعبه‌ها باشد متمایز می‌سازد. مثلاً اگر n=5, m=2, k=2 و $a_1 = 1$ $a_2 = 1$ ، $a_3 = 1$ ، $a_4 = 2$ ، $a_5 = 1$ باشد بدون در نظر گرفتن شرط مذکور ۴ شئی را می‌توان در جعبه‌ها قرار داد اما با رعایت شرطِ مسئله، تنها ۲ شئی را می‌توان جایگذاری کرد.

برای ارتباط این مسئله با مسئله ترافیک بدون در نظر گرفتن شرط « حتماً تمام شیءهای j تا n را در جعبه‌ای قرار داده باشیم » می‌توان چنین فرض کرد که در پایانه اتوبوسرانی، اتوبوس‌ها، جعبه‌ها و اشیا، مسافران هستند و ما می خواهیم که اتوبوس‌ها را با بیشترین افراد جایگذاری کنیم. گرچه این موضوع از نظر مسئله ترافیک جزء کوچکترین مسائل است، اما برای بهینه‌سازیِ حل مسئله می‌توان از روش‌هایی مانند الگوریتم‌های تکاملی و جستجوها نیز استفاده نمود. در ضمن حتی با شبکه عصبی نیز قابل یادگیری می‌باشد.

به سه حالت متفاوت مسئله را حل کرده‌ایم:

۱) متغیرهای m, n, k, a_i از قبل در برنامه مشخص باشند (فایل‌های Examples1.py and Example2.py).

۲) متغیرهای m, n, k, a_i همه توسط کاربر input شوند (فایل main.py).

۳) متغیرهای m, n, k توسط کاربر input شوند و متغیرهای a_i توسط کامپیوتر تعیین شوند (فایل Example3.py).

نام متغیرها	شرح متغیرها
n	تعداد اشیاء
m	تعداد جعبه‌ها
k	اندازه جعبه‌ها
index	محل درایه آرایه‌ی size_of_object بطوریکه $1 \leq index \leq n$ همان j در سوال است
size_of_object	آرایه ای که اندازه هر یک از اشیاء در درایه‌های آن قرار می‌گیرند (a_i)
counter_of_object	تعداد اشیاء قرار گرفته در جعبه‌ها را می‌شمارد
sum_of_sizes	مجموع اندازه اشیاء قرار گرفته در یک جعبه که باید کمتر از k باشد

سیستم عامل مورد استفاده: Windows 10

تکنولوژی و زبان مورد استفاده: نرم‌افزار PyCharm و زبان Python 3.7

دیتا بیس: ندارد

برای اجرای کد می‌توان از تمام مفسرهای زبان پایتون استفاده کرد.

لینک github: https://github.com/AZUR-lajevardi/test

Some outputs

ورودی				خروجی	
n	**m**	**k**	**a**	**j**	**بیشترین تعداد اشیا**
5	2	6	[1, 4, 1, 2, 2]	1	5
10	4	7	[7, 1, 5, 7, 6, 6, 4, 4, 6, 4]	7	4
5	2	6	[3, 4, 1, 3, 2]	2	4
5	2	3	[2, 1, 3, 1, 2]	3	3
5	5	5	[5, 5, 5, 5, 5]	1	5