



به نام خداوند جان و خرد

نام و نام خانوادگی: محمد حسین کریمی

شماره دانشجویی: 9731833

شرکت پست

تابستان 1400

توضیح کلی

این مسئله np complete است و در زمان چند جمله ای قادر به حل آن نیستیم یک راه کلی برای حل این مسائل آن است که ضریبی از راه حل بهینه را به ما می دهد که در بدترین حالت دو برابر آن است.

توضیح برنامه

در ابتدا قطعات را مرتب می کنیم سپس از سنگین ترین قطعه شروع به پیاده سازی الگوریتم زیر می کنیم:

همه ی کامیون ها را با وزن خالی در یک صف اولویت قرار می دهیم. از بین کامیون های موجود کامیونی را که بیشترین فضای خالی را دارد را انتخاب می کنیم و هر بار بزرگ ترین قطعه را در نظر می گیریم. (اگر دو کامیون بارشون برابر بود یکی را بصورت رندوم انتخاب می کنیم) و قطعه فعلی را داخل این کامیون قرار می دهیم و به سراغ قطعه ی بعدی در لیست مرتب شده می رویم (بزرگ ترین قطعه ی موجود بعدی) این الگوریتم را تا تمام شدن قطعات ادامه می دهیم. حال به یک توزیع از قطعات بین کامیون ها رسیده ایم.

اثبات نرخ تقریب

می خواهیم ثابت کنیم الگوریتم گفته شده نرخ تقریب 2 را دارد. از برهان خلف استفاده می کنیم.

برهان خلف:

فرض می کنیم حالتی وجود دارد که در آن الگوریتم گفته شده جوابی با تخمین بدتر (بیشتر از 2 برابر جواب اصلی) ارائه می دهد. طبق لم گفته شده آخرین قطعه ای که در کامیون a قرار می دهیم دارای حاکثر وزن $x + \theta/2$ است پس داخل کامیون حداقل $x + \theta/2$ بار وجود دارد و چون طبق الگوریتم کامیونی را انتخاب می کردیم که مجموعه بار آن در بین کامیون ها کمترین باشد پس در تمامی کامیون ها حداقل $x + \theta/2$ بار وجود دارد. پس یعنی مجموع بار ها حداقل برابر است با:

$$K(x + \theta/2) = k * x + k * \theta/2$$

اما در روش بهینه که وزن سنگین ترین کامیون در آن x بود مجموع وزن قطعات حداقل برابر است با kx که این تناقض است پس روشی که نرخ تقریب 2 را نقض کند یا به عبارتی نشان دهد که الگوریتم گفته شده تقریبی بزرگتر از 2 را می دهد وجود ندارد.

کامیونی را که بیشترین وزن را در الگوریتم گفته شده دارد را در نظر می گیریم. به این کامیون a می گوئیم که دارای وزن w است. حال طبق برهان خلف روشی برای اختصاص دادن قطعات به کامیون ها وجود دارد که نرخ تقریب 2 را نقض می کند پس اگر سنگین ترین کامیون در این روش دارای وزن x باشد برای w داریم:

$$w = 2x + \Theta$$

حال آخرین قطعه ای را که ما در الگوریتم گفته شده در a قراردادیم را در نظر می گیریم.

لم 1: ثابت می کنیم وزن قطعه گفته شده کمتر از $x + \Theta / 2$ است. چون قطعات را به ترتیب مرتب کردیم و از بزرگ ترین قطعه شروع کردیم اگر این اتفاق بیافتد وزن این قطعه از مجموع قطعاتی که در کامیون a قرار دارند بیشتر می شود که یعنی از تک تک آن ها بیشتر است ولی آن ها در لیست قبل از این قطعه بودند که یعنی وزن شان باید بیشتر باشد و این تناقض است.

پیچیدگی زمانی

مرتب کردن قطعات: $n \log n$

در هر مرحله برای پیدا کردن کامیونی که کمترین وزن را تا اینجا دارد از صف اولویت استفاده می کنیم که از set استفاده کردیم که اعضایش رو مرتب شده نگه می دارد و $add, delete, max, min$ از $O(n \log n)$ است ولی نکته اینجا است که عضو تکراری نگه نمی دارد برای همین اعضای که در آن ریختیم بصورت شماره کامیون - وزن کامیون هستند.

تعداد دفعات کار های بالا به اندازه تعداد قطعات است.

$$O(n \log n) + O(3n \log n) = O(n \log n)$$

اجرای برنامه

```
Enter number of trucks : 3
Enter number of elements : 6
Enter every element weight in one line :
1
3
5
7
9
11
Answer is : 12.0
{(12.0, 1), (12.0, 2), (12.0, 0)}

Process finished with exit code 0
```

```
Enter number of trucks : 3
Enter number of elements : 5
Enter every element weight in one line :
45
23
85
47
95
Answer is : 108.0
{(92.0, 2), (108.0, 1), (95.0, 0)}

Process finished with exit code 0
```