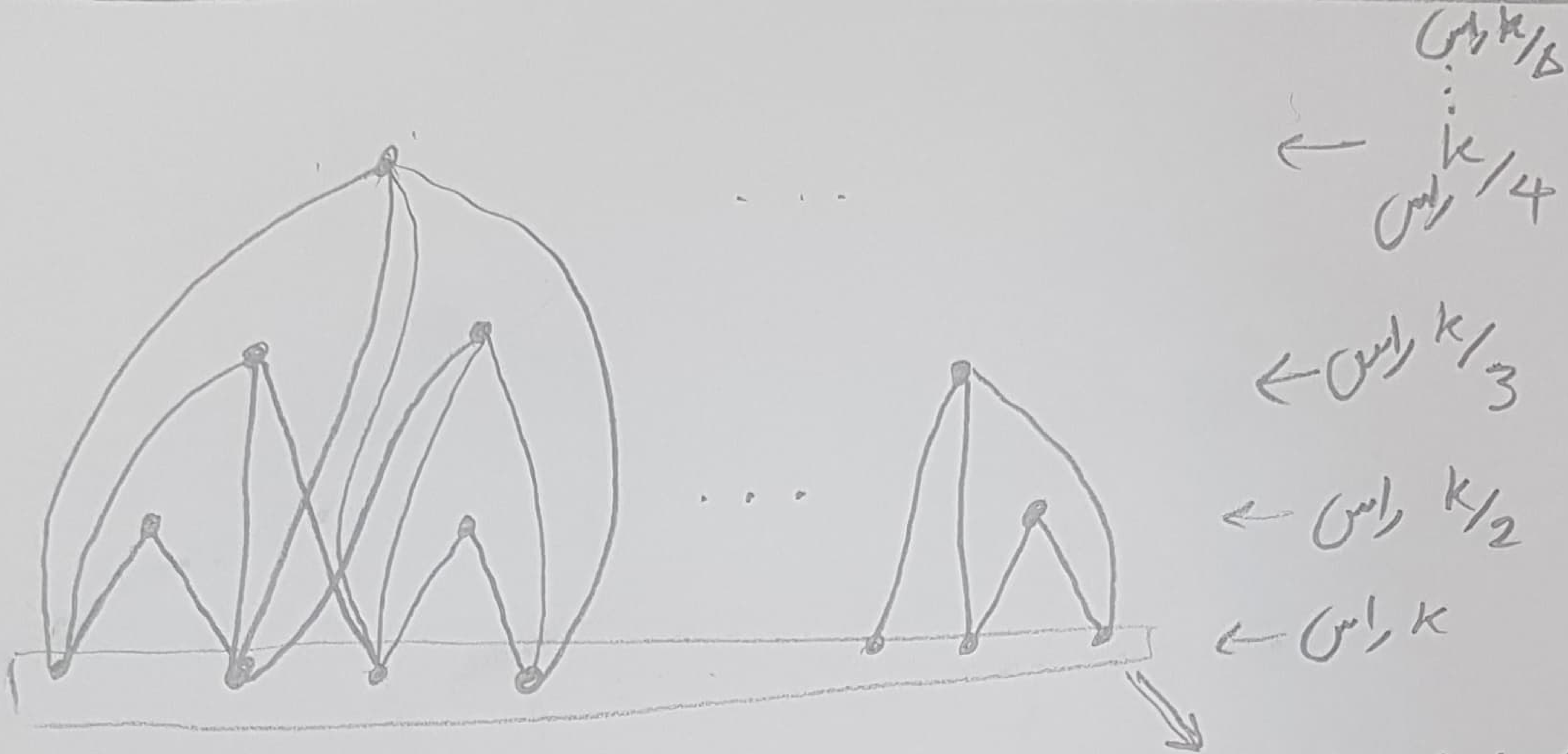
HW4 (LP_AA)
طراحی الگوریتم

مهسا کرمیان ۸۱۰۸۰۱۰۷۴
سلمان میرشعبانی ۸۱۰ ۸۰۱ ۰۸۰
معین اکبری نشهید ۸۱۰۸۰۱۰۵۵

۹



پاسخ بهینه این است که k راس پایین انتخاب شوند در حالی که الگوریتم راس با درجه بالاتر را انتخاب کرده و حذف می‌کند
این فرآیند را تکرار می‌کند:

$$A = \frac{k}{\Delta} + \cdots + \frac{k}{3} + \frac{k}{2} = k\left(\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{\Delta}\right)$$

$$\leq k\int_2^{\Delta} \frac{1}{x}\,dx = k\,[\,\mathrm{Ln}\,\Delta\,] \Rightarrow \alpha \leq \mathrm{Ln}\,\Delta \qquad A_{opt} = k$$

که $\mathrm{Ln}\,\Delta$ می‌تواند از 2 بیشتر باشد.

برای اینکه $\sum_{i=2}^{\Delta} \frac{1}{i} \geq 2$ باشد، $\Delta \geq 11$ باید باشد. اگر گراف فوق را با $\Delta \geq 11$ (یعنی راس‌های با بیش از 10 درجه) به عنوان ورودی الگوریتم بابک بدهیم، پاسخ بابک بیش از 2 برابر بدتر از پاسخ بهینه خواهد بود.

```
>>> for d in range(8, 15):
...     t = 0
...     for i in range(2, d + 1):
...         t += 1/i
...     print(d, t)
...
...

8 1.7178571428571427
9 1.828968253968254
10 1.928968253968254
11 2.019877344877345
12 2.1032106782106785
13 2.1801337551337556
14 2.2515623265623272
```

تعداد کامیون ها : m

تعداد جعبه ها : n

. W را مرتب می کنیم به طوری که $W_1 \geq W_2 \geq \dots \geq W_n$.

یک آرایه L در نظر می گیریم که سایز آن m است و میزان بار موجود در هر کامیون را نشان می دهد

در ابتدا $L_i = 0$ است. [از ۱ تا n]

بر روی W پیمایش کرده و W_j را به کامیون با کمترین بار (k) نسبت می دهیم و L آن را به روز می کنیم

پاسخ نهایی، مقدار بیشینه آرایه L خواهد بود.

$L_k += W_j$

✓ الگوریتم تمامی جعبه ها را به کامیون ها نسبت می دهد. پس به ما یک پاسخ می دهد. ☑

✓ الگوریتم در مرتبه زمانی چند جمله ای انجام می شود.

$$O(\underbrace{n\log n}_{\text{مرتب کردن } W} + \underbrace{nm}_{m \text{ بار پیدا کردن کمینه در } T})$$

← با MinHeap قابل بهبود است.

ضریب تقریب: اگر $n \leq m$ باشد، الگوریتم به هر کامیون بیش از یک جعبه نسبت نمی دهد و $A = A_{opt}$

اگر $m < n$:

پاسخ الگوریتم ← A

پاسخ بهینه ← A_{opt}

$$A_{opt} \geq \max\left(\max\{W_i\}, \frac{\sum W_i}{m}\right)$$

فرض کنیم W_k آخرین مقداری باشد که به A افزوده شده است. بار کامیون حامل A پیش از این برابر با $A - W_k$ بوده است. چون الگوریتم سبک ترین کامیون را انتخاب می کند، $m-1$ کامیون دیگر، هر کدام حداقل $A - W_k$ بار داشته اند. پس:

$$\sum_{i=1}^{n} W_i \geq (m-1)(A - W_k) + A$$

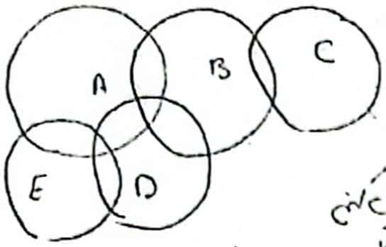
$$\Rightarrow A \leq \frac{\sum W_i}{m} + \left(1 - \frac{1}{m}\right) W_k \quad \text{I}$$

[اصل لانه کبوتری] چون m کامیون داریم و [$W_{m+1} \leq W_1, \dots, W_{m+1}$]، حداقل یک کامیون دارای دو جعبه است

پس بار آن کامیون حداقل $2W_{m+1}$ است $\Leftarrow$ $2W_{m+1} \leq A_{opt}$

واضح است که $m+1 \leq k$ $\Leftarrow$ $W_k \leq \frac{1}{2} A_{opt}$ II

$$\text{I}, \text{II} \Rightarrow A \leq A_{opt} + \left(1 - \frac{1}{m}\right)\frac{1}{2} A_{opt} \leq \frac{3}{2} A_{opt} \quad ☑$$



می‌توانیم تبدیل کنیم به لیستی که بگوید چه شهرهایی با شهرهای دیگر اشتراک دارد

" تبدیل مسئله به گراف "

$A : \{B, D, E\}$

$B : \{A, D, C\}$

$C : \{B\}$

$D : \{A, E, B\}$

$E : \{A, D\}$

زیرا تمامی اعضا داخل لیست و خود لیست باید رنگشان را دارای متفاوتی تعیین شود، می‌توان رنگشان را دارای متفاوت را به رنگ متفاوتی تشخیص کرد

" تبدیل مسئله به رنگ آمیزی گراف "

$\Rightarrow$ حالا مسئله رنگ آمیزی گراف داریم که لازم است با کمک LP حل شود.

تعریف متغیرها:

۱. می‌دانیم حداکثر k رنگ داریم (k تعداد شهرها است)، می‌توان از این k رنگ استفاده کنیم یا از آن‌ها استفاده نکنیم

$$y_k = \begin{cases} 1 & \text{اگر از } k \text{ امین رنگ استفاده کنیم} \\ 0 & \text{اگر از } k \text{ امین رنگ استفاده نکنیم} \end{cases}$$

هدف آن است که کمترین تعداد رنگ استفاده شود.

تابع هدف: کمینه‌سازی $\sum_{i=1}^{k} y_i$

2. اسم شهرها را x_i بنامیم، x_{ik} یعنی آیا i امین، رنگ k را دارد یا ندارد.

$$x_{ik} = \begin{cases} 1 & \text{ امین شهر رنگ } k \text{ ام را دارد } i \\ 0 & \text{ امین شهر رنگ } k \text{ ام را ندارد } i \end{cases}$$

مثلاً $x_{13} = 1$ یعنی اولین شهر رنگ 3 ام را گرفته، واضح است که نمی‌توانیم داشته باشیم $x_{12} = 1$, $x_{13} = 1$ چون یک شهر نمی‌تواند ۲ رنگ را بگیرد

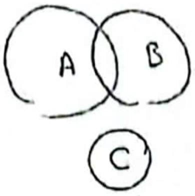
تعریف محدودیت ها :

۱. می‌دانیم هر دکل نمی‌تواند ۲ رنگ را همزمان به خود بگیرد، از طرفی هر دکل حداقل ۱ رنگ را نیز باید داشته باشد

$$\left\{ \sum_{j=1}^{k} x_{ij} = 1 \quad \text{به ازای همه‌ی رنگ‌ها} \right\}$$

← این معادله انتخاب دکل‌هاست، باید برای هر دکل معادله تکرار شود

مثلاً اگر 3 دکل A ، B ، C :



$$\left\{ \begin{array}{l} x_{11} + x_{12} + x_{13} = 1 \\ x_{21} + x_{22} + x_{23} = 1 \\ x_{31} + x_{32} + x_{33} = 1 \end{array} \right\}$$

یعنی دکل ۱ می‌تواند انتخاب کند که یکی از رنگ‌ها ۱ یا ۲ یا 3 را انتخاب کند
پس حداکثر یکی از این متغیرها ۱ باشد و بقیه باید ۰ باشند.

2. اگر رنگ kام انتخاب شود اما من در محدودیت شماره‌ی ادامه از آن استفاده کنم در این صورت تابع هدف بی‌معنا می‌شود، پس به ازای هر x_{ik} اگر x_{ik} ۱ باشد، من رنگ kام را باید قبلاً استفاده کرده باشم

(به ازای هر i و هر k)

$$y_k - x_{ik} \geq 0$$

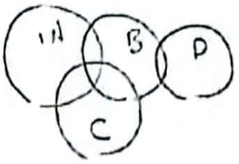
3. در صورتی که دکل‌های مجاور داشته باشیم نباید رنگ یکسان داشته باشیم مثلاً اگر دکل ۱ و ۲ مجاور هستند حق نداریم داشته باشیم

$$x_{11} = 1$$
$$x_{21} = 1$$

به ازای تمام دکل‌های مجاور هم (لیست مثال اول) و به ازای تمامی kها باید بنویسیم :

$$x_{ik} + x_{jk} \leq 1$$ (i و j شماره‌ی سطرهای مجاور است)

مثلاً :



$$\left\{ \begin{array}{l} x_{11} + x_{21} \leq 1 \\ x_{12} + x_{22} \leq 1 \\ x_{13} + x_{23} \leq 1 \\ x_{14} + x_{24} \leq 1 \end{array} \right\}$$

به ازای هر ۴ رنگ ۱، ۲، ۳، ۴ و دو دکل ۱ و ۲ :
(B ، A)

$$\left\{ \begin{array}{l} x_{11} + x_{31} \leq 1 \\ x_{12} + x_{32} \leq 1 \\ x_{13} + x_{33} \leq 1 \\ x_{14} + x_{34} \leq 1 \end{array} \right\}$$

۴. اگر مبداهای مختلف یا ستاره تامین کننده را با i و مقصدهای هر ستاره دریافت کننده را با j نمایش دهیم

هدف ما آن است که C_{ij} (Carry - Cost از مبدا i به مقصد j باید مینیمم باشد)

تابع هدف:

$$\text{minimum} \quad \sum C_{ij} \times x_{ij}$$

C_{ij} میزان هزینه‌ای است که به ازای هر واحد مواد که از ستاره i به j می‌رود محاسبه می‌شود و x_{ij} میزان تغذیه‌ای است که از ستاره i به j انتقال پیدا می‌کند.

محدودیت (۱) : (Supply)

ستاره‌های تامین کننده‌ها نمی‌توانند بیشتر از مقدارهای که معین شده تغذیه ارسال کنند پس

$$\sum_{j=1}^{n} x_{ij} \leq S_i$$ = (Supply : میزان تغذیه)

ستاره‌ی i ام ← n تا ستاره (۱ تا n) تغذیه ارسال که جمع میزان ارسال از S_i (میزانی که می‌تواند ارسال کند نباید بیشتر شود)

محدودیت (۲) (Demand)

ستاره‌ها دریافت کننده نباید کمتر از مقدار گفته شده مواد دریافت کنند (فرض می‌کنیم $S_{Total} > D_{Total}$)

$$\sum_{i=1}^{n} x_{ij} \geq D_j$$

به ازای تمام مقصدهای که مقصد از آن‌ها تغذیه دریافت می‌کنند نباید کمتر از مقدار گفته شده تغذیه دریافت شود

محدودیت (3) (درست محدودیت تابع دو شکل می‌شود)

$$x_{ij} \geq 0$$