به نام خدا

تمرین سری سوم

طراحی الگوریتم‌های پیشرفته

نعیم اصفهانی
۸۴۲۰۱۰۰۳

32.4-4

تنها فرق π' با π در ضابطه‌ی دوم تابع است. برای توضیح شباهت این دو تابع کافی است همین گزینه را بحث کنیم:

فرق گزینه‌ی دوم این است که اگر حرف بعدی در زیر الگوی انتخاب شده توسط π با حرف بعدی در زیر الگوی فعلی یکی است به صورت بازگشتی کار خود را ادامه می‌دهد. و همانند این است که یک مرحله‌ی دیگر از الگوریتم KMP اجرا شود. اگر حلقه‌ی بعد اجرا شود و شرط مذکور برقرار باشد به‌دلیل رد شدن قبلی این زیر الگو نیز رد می‌شود بنابراین این عمل تغییری در الگوریتم نمی‌دهد و جلوی تکرار حلقه را در شرایط مذکور گرفته و باعث بهبود الگوریتم می‌شود.

32-1

a.

در هر مرحله از الگوریتم KMP داریم:

$$\begin{aligned}
&if([q-\pi(q)]*\rho(\pi(q)) == \pi(q))\\
&\quad \rho(q)=\rho(\pi(q))+1\\
&else\\
&\quad \rho(q)=0
\end{aligned}$$

از آنجا که هزینه‌ی به روز نگهداری این مقدار در هر دور حلقه‌ی الگوریتم KMP، $O(1)$ است بنابراین هزینه‌ی ساختن و نگهداری این مقدار در کل برابر هزینه‌ی KMP است.

b.

$$\begin{aligned}
E(\rho^*(P)) &= \sum\left[i*2^i*\left(\frac{1}{2^{\frac{m}{i}}}\right)^i\right] = \sum\left[\frac{i*2^i}{2^m}\right] = \frac{1}{2^m}*\sum\left[i*2^i\right]\\
&\le \frac{1}{2^m}*\int_1^{m-1}\left[i*2^i*di\right] \le \frac{1}{2^m}*\left[2^{i+1}\right]_1^{m-1} \le \frac{2^m-2}{2^m}\\
&= O(1)
\end{aligned}$$

c.

بدیهی است که در صوزت اعلام یک شیفت انطباق صورت گرفته، برای اثبات درستی کافی است نشان دهیم که همه‌ی شیفت‌ها اعلام می‌شوند. تنها جایی که ما اطلاعات قبلی را از دست می‌دهیم در خط ۱۳ الگوریتم است و شرط آن عدم انطباق آخرین حرف و یا انطباق کامل است، قبل از این عمل یک عمل شیفت انجام می‌شود. این

عمل به اندازه‌ی حداقل انجام می‌شود یعنی یا یک است یا حداقل حرکت ممکن بیشتر از یک؛ در اثر $\lceil q/k \rceil$ شیفت هیچ اطلاعی از دست نمی‌رود چون k برابر $1+\rho^*(P)$ است که طبق تعریف بزرگترین تعداد مکرر بودن یک زیر الگوست. بنابراین از تقسیم q بر k کوچکترین واحد در هر حالت که می‌تواند تکراری باشد را بدست آورده پس می‌توانیم از آن بپریم.

در مورد زمان اجرای الگوریتم هم ما همیشه یک مقایسه داریم در خط ۷ و مجموع این مقایسه‌ها برابر است با $\sum q$ با توجه به شروط مربوط به q در هر لحظه؛ سعی در دقیق کردن این شروط نداریم چون $\sum \lceil q/k \rceil$ هم همان شروط را دارد و با توجه به شیفت‌ها و بیشینه گیری داریم:

$$\sum \lceil q/k \rceil \le n \Rightarrow \sum \lfloor (q+k)/k \rfloor \le n \Rightarrow \sum \lfloor q+k \rfloor \le n*k \Rightarrow \sum q = O(n*k)$$
$$= O(n*\rho^*(P)) \overset{\text{like before}}{=} O(n*\rho^*(P)+m)$$

29.2-3

از آنجا که کوتاه‌تر شدن فاصله‌ی یک راس تا مبدا تاثیر معکوس روی فاصله‌ی رئوس دیگر ندارد می‌توان به سادگی هدف کمینگی را مجموع قرار داد.

minimize:

$$\sum_{v \in V} d[v]$$

subject to:

$$d[v] \le d[u] + w(u,v) : (u,v) \in E$$
$$d[s] = 0$$

29.2-7

در اینجا کافی است به برنامه‌ریزی خطی خود مساله تابع هدف را اضافه کنیم:

minimize:

$$\sum_{u,v \in V} a(u,v) \sum_{i=1}^{k} f_i(u,v)$$

subject to:

$$\sum_{i=1}^{k} f_i(u,v) \le c(u,v) : u,v \in V$$
$$f_i(u,v) = -f_i(v,u) : i = 1 \ldots k \quad u,v \in V$$
$$\sum_{v \in V} f_i(u,v) = 0 : i = 1 \ldots k \quad u \in V - \{s_i, t_i\}$$
$$\sum_{v \in V} f_i(s,v) = d_i : i = 1 \ldots k$$

29.3-6

$$
\begin{aligned}
z &= 0 - x_1 - x_2 - x_3 \\
x_4 &= -10000 + 2x_1 + 7.5x_2 + 3x_3 \\
x_5 &= -30000 + 20x_1 + 5x_2 + 10x_3
\end{aligned}
$$

$$x_2 \Rightarrow x_5$$

$$
\begin{aligned}
z &= -6000 + 3x_1 + x_3 - 0.2x_5 \\
x_4 &= 35000 - 28x_1 - 12x_3 + 1.5x_5 \\
x_2 &= 6000 - 4x_1 - 2x_3 + 0.2x_5
\end{aligned}
$$

$$x_1 \Rightarrow x_4$$

$$
\begin{aligned}
z &= -2250 - \frac{2}{7}x_3 - \frac{3}{28}x_4 - \frac{11}{280}x_5 \\
x_1 &= 1250 - \frac{3}{7}x_3 - \frac{1}{28}x_4 + \frac{3}{56}x_5 \\
x_2 &= 1000 - \frac{2}{7}x_3 + \frac{1}{7}x_4 - \frac{1}{70}x_5
\end{aligned}
$$

$$x_1 = 1250, x_2 = 1000, x_3 = 0$$

29-1

a.

بسیار بدیهی است. با توجه به روش‌های تبدیل به فرم استاندارد ابتدا از نا مساوی‌ها فرم استاندارد را می‌سازیم و با دادن تابع هدف 0 به آن یک مساله‌ی برنامه‌ریزی خطی داریم.

b.

باید هدف را به صورت نامساوی $\sum C^T x \geq y$ درآورد. مقدار y را نیز با یک جستجوی دودویی از بیشینه تا کمینه تغییر داد با هدف پیدا کردن بیشینه مقدار y هربار الگوریتم را برای مقدار مورد نظر اجرا کرد. عملا این یک الگوریتم تقریبی است که با آن می‌شود مقدار قطعی را بدست آورد.

29-3

a.

بسیار بدیهی است چون در اثبات آن لم هرگز از فرضی مبنی بر حقیقی بودن عدد نداشتیم.

b.

با توجه به محدودیت جدید که صحیح بودن مقادیر متغیرهاست فرض الگوریتم Simplex مبنی بر صفر شدن یکی از متغیرهای پایه در هر مرحله، همیشه برقرار نیست پس در اثبات دوگانگی قوی دچار مشکل می‌شویم.

c.

بدیهی است که با توجه به محدودتر شدن، بهینگی جواب برنامه‌ریزی صحیح از برنامه‌ریزی خطی در حالت کلی کمتر است پس در مورد مساله‌ی اصلی که هدف بیشینه کردن است: $IP \le P$ و در مساله‌ی دوگان که هدف کمینه کردن است: $D \le ID$. با توجه به قاعده‌ی دوگانی قوی هم داریم: $P = D$ پس از در کنار هم گذاشتن این چند حقیقت داریم:

$$IP \le P = D \le ID$$