آنالیز الگوریتم‌ها (۲۲۸۹۱)

[بهار ۹۹]

پاسخ آزمون میان‌ترم — زمان: ۳/۵ ساعت

نکته ۱: برای مسئله‌هایی که با برنامه‌ریزی پویا حل می‌کنید لازم است زیرمسئله‌هایی که تعریف می‌کنید را به‌طور کامل و شفاف به فارسی توصیف کنید.

نکته ۲: در مسائلی که زمان اجرای الگوریتم خواسته شده مشخص شده، با ارائه الگوریتمی با زمان اجرای اندکی بدتر ممکن است بخشی از نمره را بگیرید.

۱. فرض کنید $2n$ عدد $a_1^0, a_2^0, \ldots, a_n^0$ و $a_1^1, a_2^1, \ldots, a_n^1$ داده شده باشند. می‌خواهیم یک زیرمجموعه $S = \{x_1, \ldots, x_k\} \subseteq \{1, 2, \ldots, n\}$ و رشته $b \in \{0, 1\}^k$ را بیابیم به‌طوری که برای هر i، b_i و b_{i+1} متفاوت باشند و $\sum_{i=1}^{k} a_{x_i}^{b_i}$ بیشینه شود. الگوریتمی با زمان اجرای $O(n)$ برای این مسئله ارائه دهید.

پاسخ: زیرمسئله $D(i,t)$ که $1 \leq i \leq n$ و $t \in \{0,1\}$ را برابر با جواب بهینه مسئله در حالتی که بخواهیم یک زیرمجموعه از بین i تا a^0ها و a^1های اول انتخاب کنیم و آخرین عضو انتخاب شده برابر با یک a_j^t باشد (که $j \leq i$) تعریف می‌کنیم. داریم

$$D(i,t) = \max\{D(i-1,t), D(i-1,1-t) + a_i^t\}.$$

۲. در یک درخت ریشه‌دار، منظور از یک مسیر *پایین‌رونده* مسیری مانند $v_1, v_2, \ldots, v_{k+1}$ است به‌طوری‌که برای هر $1 \leq i \leq k$، v_i پدر v_{i+1} باشد. همچنین منظور از دو مسیر مجزا، دو مسیر است که در هیچ رأسی با هم مشترک نباشند.

الگوریتمی کارآ ارائه دهید که درخت ریشه‌دار T و عدد k را ورودی بگیرد و بیشینه تعداد مسیرهای مجزای پایین‌رونده به طول k در T را محاسبه کند.

پاسخ: زیر مسئله $C(v,j)$ که $1 \leq j \leq k+1$ را برابر با جواب بهینه برای زیردرخت به ریشه v در حالتی که رأس v رأس jام از یک مسیر پایین‌رونده باشد در نظر بگیرید. جواب مسئله اصلی $C(r,1)$ است که r ریشه T است (توجه کنید می‌توانیم فرض کنیم در جواب بهینه یکی از مسیرها ریشه را شامل است). داریم

$$C(v,i) = \begin{cases} 1 + \sum\limits_{u \in v.children} C(u,1) & i = k+1 \\ \max\limits_{x \in v.children} \{ \sum\limits_{\substack{u \in v.children \\ u \neq x}} C(u,1) + C(x,i+1)\} & i \leq k \end{cases}$$

با یک‌بار محاسبه $\sum\limits_{u \in v.children} C(u,1)$ می‌توان $C(v,i)$ را در زمان $O(|v.children|)$ محاسبه کرد و بنابراین یک الگوریتم $O(nk)$ برای مسئله داریم.

۳. فرض کنید $G = (V,E)$ یک گراف بدون جهت باشد و $s \in V$. یک یال $e \in E$ را *بی‌فایده* می‌گوییم اگر با حذف e طول کوتاه‌ترین مسیر تا هر رأس $v \in V$ یا تغییر نکند یا حداکثر یکی زیاد شود. الگوریتمی با زمان اجرای $O(E)$ برای یافتن همه یال‌های بی‌فایده ارائه کنید.

پاسخ: با یک بار اجرای BFS از رأس s، $d(u)$ و $p(u)$ یعنی فاصله s تا رأس u و پدر u در درخت BFS را به ازای هر u می‌یابیم. واضح است که هر یال که به صورت $(u,p(u))$ به ازای یک u نباشد بی‌فایده است. همچنین در صورتی که به رأس u رأس $v \neq P(u)$ که $d(v) \leq d(u)$ وصل باشد، یال $(u,P(u))$ هم بی‌فایده است؛ وگرنه نیست.

۴. فرض کنید گراف $G = (V, E)$ رأسی به‌صورت ضمنی به ما داده شده باشد. مجموعه رأس‌های V برابر است با کل رشته‌های صفر و یک به‌طول n. همچنین یک تابع isConnected در اختیار داریم که می‌توانیم آن را هر دو رأس دلخواه از G مثل u و v فراخوانی کنیم و تابع به ما جواب می‌دهد که آیا u به v وصل است یا خیر. همچنین فرض کنید دو رأس $s, t \in V$ هم به ما داده شده باشند. الگوریتمی با حافظه چندجمله‌ای طراحی کنید که تشخیص دهد آیا از s به t مسیر وجود دارد یا خیر.

راهنمایی: حداکثر فاصله s و t چقدر است؟ تابعی بازگشتی بنویسید که عمق درخت بازگشتش حداکثر n باشد. توجه کنید زمان الگوریتمتان لازم نیست چندجمله‌ای باشد.

پاسخ: $\textsc{ExistsPath}(s, t, n)$ را فراخوانی می‌کنیم.

```
ExistsPath(u, v, k)
// returns True iff there is a path of length at most 2^k from u to v
if k == 0
    return isConnected(u, v)
for each x ∈ {0,1}^n
    if ExistsPath(u, x, k − 1) and ExistsPath(x, v, k − 1)
        return True
return False
```

۵. فرض کنید f یک جریان از s به t با اندازه α در شبکه G باشد. شبکه G' را به‌این‌گونه از روی G می‌سازیم که رأس‌ها و یال‌های G' همان رأس‌ها و یال‌های G هستند و ظرفیت و کران پایین برای هر یال e، به ترتیب $\lceil f(e) \rceil$ و $\lfloor f(e) \rfloor$ است. نشان دهید جریان معتبر f' در G' با اندازه $\lceil \alpha \rceil$ وجود دارد.

پاسخ: اگر یال ts با جریان α به شبکه اضافه کنیم، یک گردش داریم. پس اگر ظرفیت و کران پایین هر یال e را برابر با $f(e)$ بگذاریم، یک گردش معتبر داریم. در نتیجه این ظرفیت‌ها و کران‌پایین‌ها شرط قضیه هافمن را دارند. بنابراین اگر ظرفیت‌ها و کران‌پایین‌ها را طبق صورت سوال تعریف کنیم، همچنان شرط هافمن را داریم. حال اگر ظرفیت و کران پایین ts را از α به $\lceil \alpha \rceil$ تغییر دهیم، چون تغییر اتفاق افتاده کمتر از ۱ است و بقیه ظرفیت‌ها و کران‌پایین‌ها صحیح هستند، شرط هافمن همچنان برقرار است.

۶. گراف $G = (V, E)$ با تابع ظرفیت $c : E \to Q_+$ روی یال‌های آن را در نظر بگیرید. فرض کنید H یک گراف کامل روی V باشد. برای هر یال xy از H، $w(xy)$ را برابر با ظرفیت یک xyـ‌برش کمینه در G (برشی که x در یک سمت آن است و y در سمت دیگر) تعریف کنید.

(آ) ثابت کنید برای هر $x, y, z \in V$، $w(xy) \geq \min\{w(xz), w(zy)\}$.

(ب) فرض کنید T یک زیردرخت فراگیر از H با وزن بیشینه باشد. نشان دهید برای هر $x, y \in V$، $w(xy)$ برابر است با وزن کوچکترین یال در مسیر بین x و y در T.

پاسخ:

(آ) یک xyـ‌برش کمینه (A, B) را در نظر بگیرید. اگر $z \in A$، این برش یک zyـ‌برش هم هست و بنابراین $w(zy) \leq w(xy)$. به‌طور مشابه اگر $z \in B$، داریم $w(xz) \leq w(xy)$. پس در هر حال $w(xy) \geq \min\{w(xz), w(zy)\}$.

(ب) با استقرا روی طول مسیر می‌توان ثابت کرد که برای مسیر P از x به y، $w(xy) \geq \min_{e \in P} w(e)$. در صورتی که $P \subseteq T$، و $w(xy) > \min_{e \in P} w(e)$، می‌توان با اضافه کردن xy به T و حذف کوچکترین یال P درخت بزرگتری به دست آورد.