ساختمان داده‌ها و الگوریتم‌ها

نیم‌سال دوم ۹۷ـ۹۸

دانشکده‌ی مهندسی کامپیوتر

گردآورندگان: محمدرضا احمدخانی، عرفان فرهادی

پاسخ تمرین اول تحلیل پیچیدگی، تقسیم و حل، مرتب‌سازی

تحلیل پیچیدگی

مسئله‌ی ۱. اثبات

عبارت‌های زیر را اثبات کنید

الف) $2^{2^{2^{1000}}} \in \Theta(1)$

حل.

c را برابر خود عبارت $2^{2^{2^{1000}}}$ در نظر می‌گیریم $\triangleright$

ب) $4^n \notin O(2^n)$

حل.

فرض کنیم که $4^n \in O(2^n)$ در این صورت c و n_0ی وجود دارد که برای هر $n_0 \leqslant n$ داریم:
$4^n \leqslant c2^n \Rightarrow 2^n * 2^n \leqslant c2^n \Rightarrow 2^n < c$ که در این صورت c نمی‌تواند ثابت باشد. $\triangleright$

ج) $lg(5^x) \in \theta(x)$

حل.

$log(5^x) = xlog(5) \Rightarrow c = lg(5)$

$\triangleright$

د) $log_c(n) \in \Theta(log_2(n))$

حل.

$log_c(n) = log_2(n)/log_2(c)$
$1/log_2(c)$ ثابت است. $\triangleright$

ه) $\sum_{i=1}^{n} \frac{1}{i} \in \Theta(lgn)$

حل.

$\sum_{i=1}^{n} \frac{1}{i} \leqslant \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + ... \approx \lceil log(n) \rceil * 1$

$\sum_{i=1}^{n} \frac{1}{i} \leqslant \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + ... \approx \lceil log(n) \rceil * \frac{1}{2}$

$\triangleright$

و) $\lg(n!) \in \Theta(nlgn)$

حل.

$log(n!) = \sum_{i=1}^{n} log(i)$

$\sum_{i=1}^{n} log(i) \leqslant \sum_{i=1}^{n} log(n) = nlog(n)$

$\sum_{i=1}^{n} log(i) \geqslant \sum_{i=\frac{n}{2}}^{n} log(i) \geqslant \sum_{i=\frac{n}{2}}^{n} log(\frac{n}{2}) = \frac{n}{2} log(\frac{n}{2})$

$\triangleright$

ز) $n^k \in O(c^n)$

حل.

$$\lim_{n\to\infty} \frac{n^k}{c^n} \overset{HopitalRule}{=} \lim_{n\to\infty} \frac{kn^{k-1}}{ln(c)c^n} \overset{HopitalRule}{=} ... \lim_{n\to\infty} \frac{0}{Cc^n} = 0$$

C = یک عدد ثابت

$\triangleright$

مسئله‌ی ۲. مرتب‌سازی توابع یا توابع مرتب‌سازی. مسئله این است.

توابع زیر را به شکلی مرتب کنید که $g_i \in \Omega(g_{i+1})$

c	$n!$	4^{lgn}	e^n	$n2^n$
$\binom{n}{100}$	$(lgn)^{lgn}$	2^{2^n}	$(\sqrt{2})^{lgn}$	n^3

حل.

$$۲^{۲^n} \in \Omega(n!) \in \Omega(e^n) \in \Omega(n۲^n) \in \Omega(lg(n)^{lg(n)}) \in \Omega(\binom{n}{۱۰۰}) \in \Omega(n^۳) \in \Omega(۴^{log_n}) \in \Omega(\sqrt{۲})^{logn} \in \Omega(c)$$

$\triangleright$

مسئله‌ی ۳. ص/غ

درستی یا نادرستی عبارات زیر را مشخص کنید. (برای موارد درست اثبات و برای موارد نادرست مثال نقض بیاورید.)

الف) $f(n) \notin O(g(n)) \Rightarrow g(n) \in O(f(n))$

حل.

نادرست است. به مثال نقض توجه کنید:

$$f(n) = \begin{cases} ۱ & n = ۲k \\ n & n = ۲k + ۱ \end{cases} \quad , g(n) = \begin{cases} n & n = ۲k \\ ۱ & n = ۲k + ۱ \end{cases}$$

$\triangleright$

ب) $f(n) \in O(g(n)) \Rightarrow g(n) \in \Omega(f(n))$

حل.

اثبات: می‌دانیم $۰ \leqslant f(n) \leqslant cg(n)$ حال باید نشان دهیم که $۰ \leqslant df(n) \leqslant g(n)$ که با در نظر گرفتن $d = \frac{۱}{c}$ ثابت می‌شود. $\triangleright$

ج) $f(n) \in O(g(n)) \Rightarrow ۲^{f(n)} \in O(۲^{g(n)})$

حل.

نادرست است. به مثال نقض توجه کنید:
$۲n = O(n)$
اما $۲^{۲n} = ۴^n \notin O(۲^n)$

$\triangleright$

د) $f(n) + o(f(n)) \in \Theta(f(n))$

حل.

اثبات: فرض کنیم $g(n) = o(f(n))$ باید نشان دهیم:

$$\exists c_۱, c_۲, n_۰ : \forall n \geqslant n_۰, ۰ \leqslant c_۱ f(n) \leqslant f(n) + g(n) \leqslant c_۲ f(n)$$

با انتخاب $c_۱ = ۱$ و $c_۲ = ۲$ اثبات تکمیل می‌شود $\triangleright$

ه) $f(n) + g(n) \in \Theta(min[f(n), g(n)])$

حل.

نادرست است: به مثال نقض توجه کنید.
$n^۲ + n \neq \Theta(\min[n^۲, n]) = \Theta(n)$ $\triangleright$

و) $f(n) \in O(g(n)), \exists n_۰ : \forall n \geqslant n_۰, lg(g(n)) \geqslant ۱, lg(f(n)) \geqslant ۱ \Rightarrow$
$lg(f(n)) \in O(lg(g(n)))$

حل.

اثبات: به دلیل این‌که $f(n) \geqslant ۱$ داریم:
$\exists c, n_۰ : \forall n \geqslant n_۰, ۰ \leqslant f(n) \leqslant cg(n)$
$\Rightarrow ۰ \leqslant lgf(n) \leqslant lg(cg(n)) = lgc + lgg(n)$
باید نشان دهیم $lgf(n) \leqslant dlgg(n)$
$d = \frac{lgc+lgg(n)}{lgg(n)} = \frac{lgc}{lgg(n)} + ۱ \leqslant lgc + ۱$
گام نهایی به دلیل این‌که $lgg(n) \geqslant ۱$ برقرار است؛ پذیرفتنی‌ست $\triangleright$

مسئله‌ی ۴. توابع بازگشتی

پیچیدگی توابع بازگشتی زیر را تحلیل کنید.

الف) $T(n) = T(n - ۱) + n$

حل.

حدس می‌زنیم که $T(n) \leqslant cn^۲$ برای یک ثابتی مانند $c > ۰$ حال داریم:
$T(n) = T(n - ۱) + n \leqslant c(n - ۱)^۲ + n = cn^۲ - ۲cn + c + n = cn^۲ + c(۱ - ۲n) + n$
عبارت آخر کوچک‌تر یا مساوی $cn^۲$ است اگر $c(۱ - ۲n) + n \leqslant ۰$ یا معادلا $c \geqslant \frac{n}{۲n-۱}$.
این شرط برای همهٔ $n \geqslant ۱, c \geqslant ۱$ برقرار است. $\triangleright$

ب) $T(n) = T(\frac{n}{۲}) + \frac{n}{lgn}$

حل.

از قضیهٔ اصلی استفاده کنید:
$T(n) = \Theta(\frac{n}{logn})$ $\triangleright$

ج) $T(n) = ۲T(\frac{n}{۲}) + \frac{n}{lgn}$

حل.

از تغییر متغیر استفاده کنید: $n = 2^k, T(n) = \Theta(nlg(lg(n)))$ $\triangleright$

د) $T(n) = \sqrt{n}T(\frac{n}{2}) + 1$

حل.

از تغییر متغیر استفاده کنید: $T(n) = n^{\frac{lgn+1}{4}}$ $\triangleright$

ه) $T(n) = 2T(\frac{n}{4}) + 1$

حل.

با استفاده از قضیهٔ اصلی:

$$\theta(n^{log_4 2}) = \Theta(\sqrt{n})$$

$\triangleright$

و) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

حل.

با استفاده از قضیهٔ اصلی:

$$\theta(n^{log_4 2}lgn) = \Theta(\sqrt{n}lgn)$$

$\triangleright$

ز) $T(n) = \sum_{i=1}^{k} T(\alpha_i n) + O(n); \quad k > 1, \ \sum_{i=1}^{k} \alpha_i = 1$

حل.

حدس و استقرا:
$T(n) = \Theta(nlgn)$ $\triangleright$

مرتب‌سازی

مسئله‌ی ۵. دنباله مرتب

d دنباله از عناصر ورودی داریم. هر یک از دنباله‌ها از پیش مرتب شده هستند و تعداد کل عناصر برابر n است. الگوریتمی طراحی کنید که در $O(nlog(d))$ این d دنباله را در قالب یک دنباله مرتب کند.

حل.

در هر گام دنباله‌ها را جفت جفت در نظر بگیرید و آن‌ها را ادغام کنید. این کار را تا جایی که تنها یک دنباله باقی بماند انجام دهید. ▷

مسئله‌ی ۶. توجه به مقایسه ها

دنباله ای از اعداد صحیح داریم به گونه ای که تعداد نابه جایی‌های آن از $O(n)$ است. الگوریتم مرتب سازی مناسبی پیشنهاد کنید که تعداد مقایسه‌های آن در بدترین حالت برابر $O(n)$ باشد.

حل.

در بدترین حالت تعداد مقایسه‌های insertion sort از o(n) است. ▷

مسئله‌ی ۷. پرررررو!

n نقطه در یک دایره‌ی واحد داده شده‌اند. می‌دانیم این نقاط به صورت یکنواخت توزیع شده‌اند(احتمال یافتن یک نقطه در هر ناحیه از دایره متناسب با مساحت آن ناحیه است.). الگوریتمی پیشنهاد دهید تا نقاط را بر حسب فاصله‌ی آن‌ها از مرکز دایره مرتب کند. $\theta(n)$

حل.

دایره را به قسمت‌های مساوی تقسیم کنید و از sort bucket استفاده کنید. ▷

مسئله‌ی ۸. یعنی میشه؟

آرایه‌ی A از اعداد داده شده است.اندازه‌ی این آرایه برابر n است و هر یک از اعداد این آرایه در بازه‌ی $[۱، n^۳]$ قرار دارند. آیا میتوان این آرایه را در o(n) مرتب کرد؟

حل.

اعداد را منهای یک میکنیم. حال میتوان هر عدد را یک عدد سه رقمی در مبنای n در نظر گرفت. سپس از radix sort استفاده میکنیم. ▷

مسئله‌ی ۹. توان دو مطلوب

آرایه‌ی A از اعداد مثبت داده شده است. الگوریتمی بهینه پیشنهاد کنید تا حداکثر تعداد جفت‌هایی(j ، i) را بیابد که حاصل arr[i] + arr[j] توانی از دو باشد.
هر یک از اعداد حداکثر در یک جفت استفاده شوند.

حل.

راه حل بهینه آن است که بزرگترین عدد را در نظر بگیریم. (x) حال بزرگترین عدد آرایه را بیابیم که جمعش با x شرایط مذکور در صورت سوال را داشته باشد.
$o(nlog(n))$ ▷

مسئله‌ی ۱۰. BirdBox

سال ۲۰۵۰ است و موجودات عجیبی به جهان حمله کرده‌اند. تنها راه زنده ماندن از دست آنان پناه بردن به آرمان شهری که این موجودات قادر به ورود به آن نیستند است. برای رفتن به این آرمان شهر باید از رودخانه‌ی خروشانی عبور کنیم ولی تنها یک قایق وجود دارد که حداکثر دو نفر ظرفیت دارد. در بین راه عوارضی‌ای وجود دارد که از هر شخص هم در راه رفت و هم در راه برگشت مقداری پول می‌گیرد. چون به جز تعداد محدودی، همه خودکشی کرده‌اند مسافران مجبورند که قایق را خودشان برانند. بنابراین پس از هر سفر یکی از افراد باید برگردد تا قایق را به ساحل اولیه برساند.
n برابر با تعداد نفراتی که باید جا به جا شوند و A آرایه‌ی قیمت‌هاست. هر عنصر نشان دهنده‌ی هزینه‌ای است که هر فرد در موقعی که به تنهایی سفر کند باید بپردازد. اگر قایق دو سرنشین داشته باشد مقدار هزینه برابر بیشینه هزینه‌های انفرادی دو نفر است.
مثال:

$$n = ۴$$

$$A = [۱۰۰, ۵۰۰, ۲۰۰, ۳۰۰]$$

در این سوال حالت بهینه به این شکل است که نفر اول با نفر دوم برود و سپس نفر اول برگردد. نفر اول با نفر سوم برود و سپس نفر اول برگردد. در انتها، نفر اول با نفر چهارم برود. جمع هزینه‌ها در این حالت برابر با ۱۲۰۰ است.

الگوریتمی ارایه کنید که در $O(nlogn)$ کمترین مقدار هزینه لازم را مشخص کند.

حل.

در هر گام دو نفری که بیشترین هزینه را باید بپردازند در نظر بگیرید. برای فرستادن این دو نفر دو حالت داریم که باید بین این دو حالت مینیمم بگیریم .

حالت اول:این دو نفر با کم هزینه ترین فرد فرستاده شوند.
حالت دوم:در اولین سفر دو نفری که کمترین هزینه را دارند بروند و نفری که کمتر است برگردد. حال دو نفری که بیشترین هزینه را دارند بروند و فردی که از سفر اول فرستاده شده بود برگردد.
▷

تقسیم و حل

مسئله‌ی ۱۱. غالب ترین

در مجموعه‌ی اعداد $\{a_1, \ldots, a_n\}$عدد x را غالب می گوییم اگر بیش از نیمی از اعداد مجموعه با x برابر باشند. مجموعه‌ای از کارت‌ها داریم که پشت هر کارت یک عدد نوشته شده است. ما نمی‌توانیم اعداد پشت کارت‌ها را مشاهده کنیم، اما ماشینی وجود دارد که به ازای هر دو کارتی که به آن می دهیم به ما می‌گوید که این دو کارت برابر هستند یا خیر. می‌خواهیم بدانیم در بین اعداد پشت کارت‌ها عدد غالبی وجود دارد یا خیر و اگر وجود دارد مجموعه‌ی کارت‌هایی که این عدد پشت آن‌ها نوشته شده است را پیدا کنیم. فرض کنید تعداد کارت‌ها n باشد. الگوریتمی از مرتبه‌ی زمانی $O(nlogn)$ برای انجام این کار ارائه دهید.

حل.

از ایده‌ی تقسیم و حل استفاده کنید و سپس در هر یک از زیرمسئله‌ها حالات مختلف وجود داشتن یا نداشتن عدد غالب را ارزیابی کنید. $\triangleright$

مسئله‌ی ۱۲. یه دیواره. . .

بعد از صرف هزینه‌های میلیاردی، مسئولان دیوار جدیدی در دانشگاه ساخته‌اند که ارتفاع آن در بخش‌های مختلف،متفاوت است! در واقع این دیوار از n ستون به هم چسبیده تشکیل شده است که عرض هریک از این ستون‌ها یک متر و ارتفاع آن‌ها متفاوت است. حال مسئولان میخواهند تمام دیوار را با کاغذ رنگی بپوشانند تا به بهانه‌ی زیباسازی دانشگاه دلیلی برای این همه خرج داشته باشند. در واقع میخواهند تعدادی کاغذ رنگی به عرض یک متر به صورت افقی یا عمودی روی دیوارها بچسبانند به صورتی که تمام دیوار پوشیده شود اما هیچ دو تایی از کاغذ رنگی‌ها هم پوشانی نداشته باشند. الگوریتمی از مرتبه‌ی زمانی $O(n^2)$ ارائه دهید که کم ترین تعداد کاغذ رنگی لازم برای (پوشاندن کل دیوار را پیدا کند؟ (توجه کنید که طول کاغذ رنگی‌ها مهم نیست.

حل.

ایده مورد نظر تقسیم و حل است. در گام نخست می‌دانیم جواب n را با استفاده از کاغذ رنگی‌های عمودی میتوان به دست آورد. حال برای پیدا کردن جواب دیگر باید ابتدا اثبات شود اگر جایی کاغذ رنگی افقی نصب شده بود کاغذ رنگی‌های زیر آن هم افقی هستند. در نهایت شما در هر گام باید تا لب کوتاه ترین ستون را با کاغذ رنگی افقی پر کنید و حال سوال به دو قسمت تقسیم می‌شود. کمینه جواب به دست آمده در این قسمت و جواب قسمت قبل(n) برابر جواب کلی سوال است. $\triangleright$

موفق باشید :)