انجمن علمی مهندسی کامپیوتر
دانشگاه مازندران

به نام خدا

پاسخ تمرین اول ساختمان داده‌ها و الگوریتم

الگوریتم‌ها و حل مسئله - پیچیدگی زمانی - الگوریتم های بازگشتی

استاد:

دکتر روستایی

استادیار:

عرشیا عموزاد

۱. یادآوری: اگر $f(n)$ و $g(n)$ توابعی صعودی باشند و $\log(f(n))$ از $\log(g(n))$ رشد بیشتری داشته باشد، آنگاه $f(n)$ از $g(n)$ رشد بیشتری دارد.
البته در صورتی که رشد لگاریتم توابع از مرتبه‌ی یکسانی شود، نمی‌توان درباره هم مرتبه بودن توابع اظهار نظر کرد.

$$\log(g(n)) = \log(n^{\log n}) = \log^2 n$$

$$\log(f(n)) = \log(\log^n n) = n\log(\log(n))$$

$$\lim_{n\to\infty} \frac{\log(f(n))}{\log(g(n))} = \lim_{n\to\infty} \frac{n\log(\log(n))}{\log^2(n)}$$

$$\log n = m, n = 2^m$$

$$\lim_{n\to\infty} \frac{2^m \log(m)}{m^2} = \infty \Rightarrow g(n) = O(f(n))$$

۲. مانند سوال قبل رفتار می کنیم.

$$\log f(n) = \log n. \log 4 = 2\log n = \Theta(\log n)$$
$$\log g(n) = \log^2(n) = \Theta(\log^2 n)$$
$$\log h(n) = 2\log(\log(n)) = \Theta(\log(\log(n)))$$
$$\Rightarrow h(n) \leq f(n) \leq g(n)$$

۳. یادآوری:

$$a^b = e^{b.\ln a}$$

طبق نکته‌ی بالا خواهیم داشت:

$$f_3(n) = f_5(n) = e^{\log n. \log\log n}$$

حال توابع را بصورت لگاریتمی با هم مقایسه می‌کنیم:

$$\ln f_1 \sim 1.01 \ln n$$
$$\ln f_2 \sim (\sqrt{n}). \ln 2$$
$$\ln f_4 \sim (lnn)^2$$
$$\ln f_3 = \ln f_5 \sim \ln n. \ln\ln n$$

با مقایسه لگاریتم توابع، ترتیب صعودی توابع از کوچک به بزرگ خواهد شد:

$$f_1 < f_3 \equiv f_5 < f_4 < f_2$$

۴. گزینه‌ی ۱
تعداد n قطعه گوشت خرس را n روز زنده نگه می‌دارد.
$\frac{n}{2}$ از این قطعات زوج است که این قطعات مجددا ذخیره می‌شوند که در نتیجه خرس را $\frac{n}{2}$ روز دیگر زنده نگه می‌دارد.
سپس $\frac{n}{4}$ دیگر از قطعات زوج است و باقی می‌ماند در نتیجه خرس $\frac{n}{4}$ روز دیگر نیز زنده می‌ماند. به همین ترتیب باقی روزهای عمر خرس برابر است با:

$$n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \frac{n}{16} = n(\sum_{i=0}^{\infty}) = n * \frac{1}{1-\frac{1}{2}} = 2n$$

۵. ۱. حل با استفاده از قضیه اصلی:

$$a = 4, b = 2, f(n) = \log^2 n! = \Theta(n^2 \log^2 n)$$
$$n^{\log_b a} = n^2$$
$$f(n) = \Theta(n^{\log_b a}. \log^k n) \Rightarrow T(n) = \Theta(n^{\log_b a}. \log^{k+1} n)$$
$$\Rightarrow T(n) = \Theta(n^2. \log^3 n)$$

۲. اثبات:

$$\log n! = \sum_{i=1}^{n} \log i$$
$$\sum_{i=1}^{n} \log i \leq \sum_{i=1}^{n} \log n = n \log n$$
$$\sum_{i=1}^{n} \log i \geq \sum_{i=\frac{n}{2}}^{n} \log i \geq \sum_{i=\frac{n}{2}}^{n} \log \frac{n}{2} = \frac{n}{2}. \log \frac{n}{2}$$
$$\Rightarrow \log n! = \Theta(n \log n)$$

۶. ۱. برای تقسیم های ثابت مثل نسبت ۱ به ۳ یا ۱ به ۴، بازگشت $T(n) = T(b_1 n) + T(b_2 n) + \Theta(n)$ را داریم و جواب $T(n) = \Theta(n \log n)$ است.
نسبت نامناسب فقط ضریب ثابت را بزرگ تر می‌کند زیرا عمق درخت افزایش می‌یابد، ولی ترتیب رشد $n \log n$ حفظ می‌شود.
اگر تقسیم خیلی نامتوازن باشد(مثلا یکی همیشه $n - 1$)، پیچیدگی می‌تواند به $\Theta(n^2)$ برسد.

۲. اگر کل آرایه مرتب صعودی باشد، آنگاه برای هر تقسیم متوازن، تمام عناصر نیمه چپ کوچکتر یا مساوی تمام عناصر نیمه راست هستند. بنابراین وقتی دو نیمه را با الگوریتم مرج استاندارد ادغام میکنیم، مقایسه ها به شکل همیشه برد عنصر از نیمه چپ انجام می‌شوند تا زمانی که نیمه چپ خالی شود. یعنی در هر ادغام دو نیمه با اندازه های برابر a و a تعداد مقایسه های انجام شده برابر a خواهد بود.
در سطح پایینی (ادغام زوج های تک عنصری) تعداد مقایسه ها در آن سطح برابر $\frac{n}{2}$ است.
در هر سطر بالاتر، برای هر ادغام دو زیرآرایه‌ی برابر اندازه‌ی 2^{j-1} مقایسه های انجام شده برابر 2^{j-1} است و تعداد ادغام ها در آن سطح برابر $\frac{n}{2^j}$ است. بنابراین مجموع مقایسه ها در هر سطح همیشه برابر $\frac{n}{2}$ است.
عمق درخت برابر $\log n$ است.
پس مجموع کل مقایسه ها در حالت کاملا مرتب:

$$Comparisons_{sorted} = \frac{n}{2}. \log n$$

۷. تابع آکرمان را باید تریس کنیم که خواهیم داشت:

$$func(2,3) = func(1, func(2,2)) = func(1,7) = 9$$
$$func(2,2) = func(1, func(2,1)) = func(1,5) = 7$$
$$func(2,1) = func(1, func(2,0)) = func(1,3) = 5$$
$$func(2,0) = func(1,1) = 3$$
$$func(1,1) = func(0, func(1,0)) = func(0,2) = 3$$
$$func(1,0) = func(0,1) = 2$$
$$func(0,1) = 2$$