ساختمان داده‌ها (۲۲۸۲۲)

مدرس: حسین بومری

[زمستان ۹۹]

سوال ۵: الگوریتم ضعیف تحلیل ق — نگارنده: آئیریا محمدی

رشته‌ها را به شکل مرتب شده بر اساس طول و با نماد $w_1, \cdots, w_n$ نشان می‌دهیم که n تعداد رشته ها است و طول هرکدام l_i و داریم $l_i < l_{i+1}$.

فرض کنیم m_1 پرسمان از رشته‌های جفت طول کوچک‌تر مساوی x یا یکی بزرگ‌تر و یکی کوچک‌تر از x انجام دهیم و $m_2 = m - m_1$ پرسمان با هر دو رشته با طول بیشتر .

مرتبه زمانی اجرای الگوریتم بر روی دو رشته از مرتبه طول رشته کوتاه‌تر است پس برای m_1 پرسش اول داریم $T = O(m_1 \cdot x)$.

برای m_2 پرسش نوع دوم به شکل زیر استدلال می‌کنیم :

تنها یک زوج (w_n, w_n) داریم که طول رشته کوچک‌ترش برابر با l_n است و دو زوج که طول رشته کوچک‌ترش برابر با l_{n-1} و .... هزینه پرسمان‌های نوع دو به شکل $T = l_n + 2l_{n-1} + \cdots + (y+2)l_{n-y+1}$ می‌شود که $\sum_{i=1}^{y+2} 1 = m_2$ یعنی $y \in \Theta(\sqrt{m_2})$.

رابطه بالا در صورتی بیشینه می‌شود که $l_n = l_{n-1} = \cdots = l_{n-y+1}$

و حداکثر مقدار $l_i \; (i \in [n-y+1, n])$ برابر می‌شود با وقتی که $n_1 = 0$ و $n_2 - y$ رشته به طول x و y رشته به طول $l_i = \frac{k - x(n_2 - y)}{y}$ داشته باشیم . پس $l_i \in O(\frac{k}{y} + x)$ و داریم $y \in \Theta(\sqrt{m_2})$ پس l_i به شکل $l_i \in O(\frac{k}{\sqrt{m_2}} + x)$ است .

و برای مرتبه زمانی m_2 پرسش نوع دو خواهیم داشت $T = l_i(1 + 2 + \cdots + (y+2)) \in l_i \cdot O(y^2) = O((\frac{k}{\sqrt{m_2}} + x)m_2)$

از آن‌جایی که طول تمام رشته ها (در نوع دو) از x بزرگ‌تر است و جمع طول آن‌ها حداکثر k می‌شود پس حداکثر تعداد آن‌ها $n_2 = \frac{k}{x}$ می‌باشد .

از طرفی اگر $n_2 = \frac{k}{x}$ و $n_1 = n - n_2$ را تعداد رشته‌های با طول بزرگ‌تر و کوچک‌تر از x در نظر بگیریم، در صورت وجود محدودیت برای پرسمان تکراری حداکثر $\binom{n_2}{2}$ زوج رشته از نوع دو می‌توان انتخاب کرد که از $O((n_2)^2)$ می‌باشد و مشابها تعداد زوج‌های قابل انتخاب از رشته‌های نوع اول از مرتبه $O((n_1)^2)$.

پس مجموع هزینه‌ای که برای این دو نوع پرسش خواهیم داد به شکل زیر خواهد بود :

$$T = O(x(n - n_1)^2) + O(k(n_2)^2) = O(x(n - \frac{k}{x})^2) + O((\frac{k}{\sqrt{m_2}} + x)m_2)$$

به ازای $x = \sqrt{k}$ خواهیم داشت

$$T \in O((n - \sqrt{k})^2\sqrt{k} + k\sqrt{m_2} + m_2\sqrt{k}) = O((n^2 + k + m_2)\sqrt{k} + k\sqrt{m_2}) = O((n^2 + k + n_2^2)\sqrt{k} + kn_2)$$

ب)

کافیست که پاسخ ها را ذخیره کنیم. درختی دودویی متوازن نگهداری میکنیم که هر گره آن نشان دهنده اندیس رشته اول یک پرسمان است و در آن درخت دیگری نگهداری می‌شود که هر گره نشان هنده اندیس دوم پرسمان است و محتوای ان پاسخ این پرسمان است.

از آن‌جایی که m پرسمان تنها داریم و ارتفاع هر دو درخت حداکثر از O(m) است (در هر پرسمان یک نتیجه به آن اضافه می‌شود) و جستجو در درخت از مرتبه ارتفاع است پس مرتبه زمانی $O(mlog^2m)$ زمان به زمان الگوریتم اصلی اضافه می‌شود . از طرفی $\binom{m\leq n}{2\in O(n^2)}$ پس زمان اضافه شده می‌شود $O(mlog^2n^2) = O(mlog^2n) = O(mlogK)$