

ساختمان داده‌ها و الگوریتم‌ها

پاسخ کوئیز دوم - مرتب‌سازی و درهم‌سازی

شماره دانشجویی:

نام و نام خانوادگی:

تاریخ: ۱۴۰۲/۱۰/۱۲

۴۰ نمره

۱.

یک جدول درهم‌سازی به طول $m = 11$ داریم که در ابتدا همه‌ی خانه‌های آن خالی است. با استفاده از هر کدام از روش‌های کنترل تصادم زیر، اعداد ۲۵، ۱۴، ۲۱، ۳، ۳۶ را به ترتیب از راست به چپ در جدول وارد کنید:

الف) کاوش مرتبه دو (*quadratic probing*) با $h(k, i) = (h'(k) + i + 3i^2) \bmod m$, $h'(k) = k$
ب) درهم‌سازی دوگانه (*double hashing*) با $h_1(k) = k$, $h_2(k) = 1 + (k \bmod (m - 1))$

پاسخ:

الف) با استفاده از کاوش مرتبه دو جدول hash به صورت زیر خواهد بود:

Index Step	Solution	0	1	2	3	4	5	6	7	8	9	10
Insert 25	$h(25, 0) = 3$				25							
Insert 14	$h(14, 0) = 3$ collision $h(14, 1) = 7$				25				14			
Insert 21	$h(21, 0) = 10$				25				14			21
Insert 3	$h(3, 0) = 3$ collision $h(3, 1) = 7$ collision $h(3, 2) = 6$				25			3	14			21
Insert 36	$h(36, 0) = 3$ collision $h(36, 1) = 7$ collision $h(36, 2) = 6$ collision $h(36, 3) = 0$	36			25			3	14			21
Final	-	36			25			3	14			21

ب) با استفاده از درهم‌سازی دوگانه جدول hash به صورت زیر خواهد بود:

Index Step	Solution	0	1	2	3	4	5	6	7	8	9	10
Insert 25	$h_1(25) = 3$				25							
Insert 14	$h_1(14) = 3$ collision $h_2(14) = 5$ $index = 3 + 1 \times 5 = 8$				25					14		
Insert 21	$h_1(21) = 10$				25					14		21
Insert 3	$h_1(3) = 3$ collision $h_2(3) = 4$ $index = 3 + 1 \times 4 = 7$				25				3	14		21
Insert 36	$h_1(36) = 3$ collision $h_2(36) = 7$ $index1 = 3 + 1 \times 7 = 10$ collision $index2 = 3 + 2 \times 7 = 17 \equiv 6$				25			36	3	14		21
Final	-				25			36	3	14		21

۲.

۲۰ نمره

اثبات کنید چرا در الگوریتم های مرتب سازی مقایسه ای ، پیچیدگی زمانی بهتر از $O(n \log n)$ نمیتوانیم داشته باشیم.

پاسخ:

درخت تصمیم یک روش برای معادل سازی تصمیم گیری است که گره های داخلی شرط ها و برگ های درخت انتخاب های نهایی ما هستند پس در نهایت ارتفاع این درخت برابر است با:

$$\log_2(\text{numberOfLeaves}) \leq h$$

چون برگ های درخت تصمیم ما انتخاب های نهایی ما هستند پس در نهایت $n!$ برگ خواهیم داشت.

$$h \geq \log_2(\text{NumberOfLeaves}) \geq n \log_2(n)$$

بنابراین:

$$h \geq n \log_2(n)$$

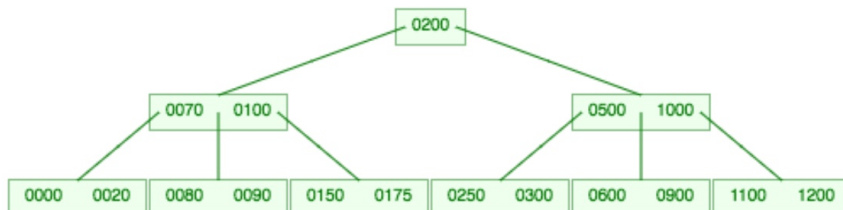
برای مطالعه بیشتر به سایت زیر مراجعه کنید .

<https://www.jmoisio.eu/en/blog/2020/09/12/why-sorting-cannot-be-done-faster-n-log-n/>

۳.

۴۰ نمره

درخت زیر از order ۵ میباشد. حالا مقدار ۱۰۰ را از BTree زیر حذف نمایید.



پاسخ:

۱۰۰ در یک گره داخلی قرار دارد، بنابراین ما بزرگترین کلید در زیردرخت چپ آن را حذف می کنیم، که ۹۰ است (این مورد همچنین می تواند به عنوان حذف جانشین بلا فصل ۱۰۰ بیان شود) و ۱۰۰ را با ۹۰ جایگزین می کنیم. سپس به بازتنظیم گره برگی که ۹۰ از آن حذف شده است، می پردازیم. از آنجا که آن گره برگی اکنون فقط کلید ۸۰ را دارد، که کمتر از حداقل تعداد کلیدهای مورد نیاز، یعنی ۲، است، ما به دنبال گرفتن کلیدی از یک گره هم سطح هستیم و می بینیم که این کار ممکن نیست زیرا هر دو گره هم سطح تعداد حداقل کلیدهای لازم را دارند. بنابراین ما باید گره ۸۰ را با گره هم سطح چپ آن که شامل کلیدهای ۲۰ و ۰ است، ادغام کنیم. هنگام انجام این ادغام، ما همچنین ۷۰ را از والد تنزل داده و در گره برگی جدید وارد می کنیم زیرا ۷۰ قبلاً به دو گره برگی جدا اشاره می کرد و اکنون آنها ادغام شده اند. بنابراین ۷۰ نیز باید در گره برگی جدید گنجانده شود، که به ما گره برگی می دهد که شامل کلیدهای ۰، ۲۰، ۷۰، و ۸۰ است.

به دلیل تنزل ۷۰، ما با یک گره داخلی مواجه هستیم که فقط شامل ۱ کلید، یعنی ۱۰۰ است. از آنجا که این کمتر از حداقل تعداد کلیدهای مورد نیاز است، ما به دو گره هم سطح آن نگاه می کنیم تا ببینیم می توانیم کلیدی را قرض بگیریم. هیچ گره هم سطح چپ وجود ندارد و گره

هم‌سطح راست حداقل تعداد کلیدهای لازم را دارد، بنابراین نمی‌توانیم کلیدی بگیریم. در عوض ما باید گره داخلی حاوی ۱۰۰ را با یک گره هم‌سطح ادغام کنیم. معمولاً ما با گره هم‌سطح چپ ادغام می‌کنیم، اما ۱۰۰ چپ‌ترین گره در این سطح است بنابراین گره هم‌سطح چپ وجود ندارد. بنابراین ما ۱۰۰ را با گره هم‌سطح راست ادغام می‌کنیم. ما همچنین کلید والد برای این دو گره، که ۲۰۰ است را تنزل می‌دهیم، و در نتیجه یک گره داخلی جدید با کلیدهای ۱۰۰، ۲۰۰، ۵۰۰ و ۱۰۰۰ به دست می‌آوریم. از آنجا که ۲۰۰ تنها کلید در والد (که گره ریشه درخت B بود) بود، والد ناپدید می‌شود و درخت B یک سطح ارتفاع کم می‌کند. گره داخلی جدید به ریشه جدید درخت B تبدیل می‌شود. درخت B حاصل به صورت زیر است:

