به نام خدا
کوئیز سوم درس ساختمان داده‌ها و الگوریتم‌ها
مدرس: مهدی صفرنژاد ـ ۴ آذر ۱۴۰۰
زمان پاسخگویی:۳۰ دقیقه زمان آپلود: ۱۵ دقیقه

سوال ۱:

الگوریتمی ارائه کنید که در زمان $O(h)$ کم‌ترین فاصله‌ی بین دو راس متمایز x ، y در درخت دودویی جستجو را محاسبه کند (h برابر ارتقاع درخت است). توضیح دهید چرا پیچیدگی زمانی الگوریتم ارائه شده برابر $O(h)$ است.

پاسخ
برای حل این سوال باید نزدیک‌ترین جد مشترک x و y را پیدا کنیم. از ریشه درخت شروع می‌کنیم. در صورتی که هر دو راس داده شده از ریشه بزرگ‌تر باشند ،فرزند راست ریشه و در صورتی که هر دو از ریشه کوچک‌تر باشند، فرزند چپ ریشه راسی خواهد بود که در ادامه بررسی خواهد شد و در صورتی که یکی از راس‌ها از ریشه بزرگ‌تر و دیگری کوچک‌تر باشد آنگاه ریشه نزدیک‌ترین جد مشترک این دو راس است. همین روند را ادامه می‌دهیم تا نزدیک‌ترین جد مشترک دو راس پیدا شود. سپس از راس پیدا شده، دو راس x و y را پیدا می‌کنیم و مسیر خواسته شده بدست می‌آید. در این الگوریتم پیدا کردن ریشه و هر راس در $O(h)$ امکان پذیر خواهد بود(برای پیدا کردن ریشه، در هر عمقی از درخت $O(1)$ عملیات صورت می‌گیرد در نتیجه در نهایت $O(h)$ عملیات داریم). در نتیجه پیچیدگی زمانی الگوریتم نیز برابر $O(h)$ خواهد بود.

سوال ۲:

امیر در اتاقی زندانی شده است و در صورتی که بتواند به سوالات پرسیده شده توسط زندان بان پاسخ بدهد مقداری گوشت به عنوان غذا دریافت می‌کند. زندان‌بان بعد از هر بار غذا دادن به امیر از او می پرسد میانه‌ی وزن گوشت هایی که تا الان دریافت کرده چه مقدار بوده است. با استفاده از داده ساختار heap الگوریتمی ارائه کنید که امیر بتواند در زمان $log(n)$ که n برابر تعداد گوشت‌هایی است که دریافت کرده، به سوال زندان‌بان پاسخ دهد.
توضیح دهید چرا پیچیدگی زمانی الگوریتم ارائه شده برابر $O(logn)$ است.

پاسخ
برای حل این سوال داده ساختار «صف اولویت میانه» را طراحی می‌کنیم. در این داده ساختار با هزینه $O(1)$ به میانه‌ی داده‌ها دسترسی داریم و با هزینه $O(logn)$ می‌توانیم عضو جدیدی به داده ساختار اضافه کنیم. در نتیجه با اضافه شدن هر وزن گوشت جدید با هزینه‌ی $O(logn)$ می‌توانیم پاسخ زندان‌بان را بدهیم.
برای پیاده‌سازی این داده ساختار، از یک هرم کمینه و یک هرم بیشینه استفاده می‌کنیم. به این صورت که، کل عناصر را از نظر اندازه به دو ناحیه‌ی تقریبا مساوی بزرگتر و کوچکتر تقسیم می‌کنیم، به طوری که از تعداد عناصر حداکثر یکی با هم تفاوت داشته باشند. حال، ناحیه‌ی بزرگتر را درون یک هرم کمینه، و ناحیه‌ی کوچکتر را در یک هرم بیشینه قرار می‌دهیم. با توجه به این پیاده‌سازی، اعمال مختلف به این صورت پیاده‌سازی می‌شوند:
درج عنصر: در صورتی که عنصر جدید از ریشه‌ی هرم کمینه بیشتر باشد درون هرم کمینه و در غیر این صورت در هرم بیشینه درج می‌شود. پس از درج درون هرم، در صورتی که تفاوت تعداد عضو هرم‌ها بیشتر از یک باشد، یک عضو از هرم با اعضای بیشتر حذف و به هرم دیگر اضافه می‌شود. چون درج در هرم و حذف از آن از $O(logn)$ است و اندازه هرم هم تقریبا $n/2$ است در نتیجه این عمل $O(logn)$ خواهد بود.
دریافت عنصر میانه:

عنصر میانه برابری ریشه‌ی هرمی است که عضو بیشتری دارد و در صورتی که هر دو هرم تعداد عضو برابری داشته باشند برابر میانگین دو ریشه خواهد بود.