

## پاسخ تشریحی توسط: جواد گرشاسبی

37. گزینه 2 درست است.

در این هرم، اجباری وجود ندارد که سومین کوچک‌ترین عنصر، فرزند ریشه باشد و می‌تواند فرزند یکی از فرزندان ریشه باشد، در نتیجه گزاره گزینه دوم نادرست است.  
گزاره‌های اول و سوم نیز صحیح هستند.

38. گزینه 4 درست است.

با یک درخت مورب با 6 گره a تا f می‌توان هر 3 گزاره اول را نقض کرد، در نتیجه گزینه هیچکدام صحیح است.

39. گزینه 4 درست است.

با توجه به جدول نهایی می‌توان درمورد ترتیب درج عناصر به نتایج زیر رسید:

عنصر C باید قبل از A درج شود.

عنصر D نیز باید قبل از A درج شود ولی می‌تواند قبل یا بعد از C باشد.

عنصر E می‌تواند قبل یا بعد از هر کدام از عناصر C، D و A درج شود ولی حتماً باید قبل از عناصر F و B باشد.

عناصر F و B باید بعد از تمام عناصر درج شوند و همچنین F باید قبل از B درج شود.

با توجه به شروط ذکر شده روش‌های ممکن برای درج عناصر 8 روش هستند و عبارتند از:

CDAEFB, CDEAFB, CEDAFB, DCAEFB, DECAFB, DCEAFB, ECDAFB, EDCAFB

40. گزینه 1 درست است.

یک درخت قرمز-سیاه با 1023 عنصر می‌تواند یک درخت پر باشد و درخت قرمز-سیاه پر می‌تواند گره قرمز رنگ نداشته باشد، چون در اینصورت نیز در مسیر ریشه تا تمام نوادگان تعداد گره‌های سیاه برابر است، در نتیجه حداقل تعداد گره‌های قرمز در این درخت قرمز - سیاه صفر گره است.

41. گزینه 2 درست است.

برای حل این سوال می‌توان عناصر مجموعه را به صورت یک لیست مرتب در نظر گرفت و با استفاده از دو اشاره‌گر i و j که به ترتیب به ابتدا و انتهای لیست اشاره می‌کنند مساله را اینگونه حل کرد که ابتدا مقادیر خانه‌های i و j را با یکدیگر جمع می‌کنیم، اگر با مقدار c برابر بود که پاسخ را یافته‌ایم، در غیر اینصورت اگر حاصلجمع کوچک‌تر از c بود، اشاره‌گر i را یک خانه به جلو می‌بریم و دوباره عمل مقایسه را انجام می‌دهیم، در غیر اینصورت و اگر حاصلجمع بزرگ‌تر از c بود، اشاره‌گر j را یک خانه به عقب

حرکت می‌دهیم و تا یافتن دو خانه که مجموع محتوای آن‌ها برابر با  $c$  باشد و یا رسیدن دو اشاره‌گر به یکدیگر به معنی عدم وجود دو عدد که مجموع آن‌ها  $c$  است، این فرایند را ادامه می‌دهیم. واضح است که در بدترین حالت به تعداد عناصر موجود در لیست، باید اعمال جمع، مقایسه و حرکت دادن اشاره‌گرها انجام شود، در نتیجه مرتبه زمانی این الگوریتم  $O(n)$  است.

#### 42. گزینه 1 درست است.

روند کار بدین صورت است که در ابتدا  $k$  امین کوچک‌ترین عنصر را در مرتبه زمانی  $O(n)$  پیدا می‌کنیم، سپس این عنصر را به عنوان محور قرار داده و عمل پارتیشن را در مرتبه زمانی  $O(n)$  انجام می‌دهیم و در آخر عناصر کوچک‌تر از این عنصر را که تعداد آن‌ها  $k$  عنصر است را با استفاده از یک الگوریتم مرتب‌سازی در مرتبه زمانی  $O(k \lg k)$  مرتب می‌کنیم، بدین ترتیب مرتبه زمانی اجرای این الگوریتم  $O(n + k \lg k)$  خواهد بود.