

# طراحی الگوریتم

مهداد صفی خانی

۱۳۹۹/شهریور

سوالات نیمسال اول ۹۶-۹۷ سوالات زوج

۲- کدام گزینه رابطه بازگشتی مسئله برج هانوی را نشان می‌دهد؟

۱.  $T(n)2^n - 1, T(n) = 2T(n-1) + 1$

۲.  $T(n) = 2^{2-1}, T(n) = 2T(n-1) + 1$

۳.  $T(n) = \log n, T(n) = 2T() + 1$

۴.  $T(n) = n + \log n, T(n) = 2T() + 1$

پاسخ:

در مسئله برج هانوی در هر بار اجرا دو بار  $n-1$  دیسک و یک بار یک دیسک جابه‌جا می‌شوند لذا برج هانوی  $T(n) = 2T(n-1) + 1$  می‌باشد اگر رابطه به صورت  $T(n) = aT(n-k) + b$  باشد مرتبه زمانی آن برابر با  $O(a^{\frac{n}{k}})$  خواهد بود لذا در گزینه ۱ که  $T(n) = 2^n - 1$  می‌باشد صحیح است.

۴- تابع بازگشتی زیر را در نظر بگیرید ( $n$  توانی از ۲ است) زمان اجرای فوق چیست؟

```
Int F(int n)
{
    if(n <= return1);
    else return(F(+));
}
```

۱.  $O(\log n)$

۲.  $O(n)$

۳.  $O(n \log n)$

۴.  $O(n) + 1$

پاسخ: الف

$$T(n) = T(\sqrt{n}) + 1 \rightarrow n = 2^m \rightarrow T(2^m) = T(2^{\frac{m}{2}}) + 1 \rightarrow T(m) = T(\frac{m}{2}) + 1$$

با توجه به قضیه اصلی:  $a=1, b=2, k=0$

$$b^k = 2^0 = 1 \rightarrow b^k = a$$

در نتیجه  $T(m) = \theta(\log m)$ ، از آنجا که  $n = 2^m \rightarrow m = \log n$  پس:

$$T(n) = \theta(\log \log n)$$

۶- جواب رابطه بازگشتی کدام یک از گزینه‌های زیر است؟

$$T(n) = 9T(n/3) + n$$

$$1. \theta(n^2 \log n)$$

$$2. \theta(n \log n)$$

$$3. \theta(n)$$

$$4. \theta(n^2)$$

پاسخ: د

از قضیه اصلی استفاده می‌نماییم:

$$T(n) = aT\left(\frac{n}{b}\right) + n^k$$

$$T(n) = 9T\left(\frac{n}{3}\right) + n \rightarrow a = 9, b = 3, k = 1$$

a با  $b^k$  مقایسه می‌شود

اگر  $a > b^k$  باشد مرتبه زمانی  $n^{\log_b a}$  خواهد بود

$$a > b^k \rightarrow T(n) = O(n^{\log_b a}) = O(n^{\log_3 9}) = O(n^2)$$

۸- الگوریتم sort Quick یک رشته n تایی را در حالت متوسط با چه سرعتی مرتب می‌کند؟

۱.  $O(n \log n)$

۲.  $O(n)$

۳.  $O(n^2)$

۴.  $O(\log n)$

پاسخ : الف

نتایج زیر را برای الگوریتم مرتب سازی سریع داریم:  
زمانی که داده ها از قبل مرتب شده باشند الگوریتم در بدترین حالت خود می باشد.  
 $O(n^2)$  بدترین حالت و  $O(\log n)$  حالت میانگین

۱۰- بدترین حالت الگوریتم های تقسیم و حل برای  $n$  ورودی کدام گزینه است؟

۱. مسئله به تعدادی زیر مسئله تقسیم شود.

۲. مسئله به قسمت های مساوی تقسیم شود

۳. مسئله به سه قسمت تقسیم شود.

۴. مسئله به  $n$  قسمت تقسیم شود.

پاسخ : د

برای مسائلی که با تقسیم مسئله اصلی، مسائل کوچکتر دوباره به اندازه تقریباً  $n$  باشد و مسائلی که به تعداد زیادی زیر مسئله با طول  $(n/c)$  تقسیم می شود، روش تقسیم و حل مناسب نیست.

۱۲- هزینه ی درخت پوشای مینیمم (Spanning (Minimum Tree) گراف زیر چیست؟

۱. ۶۸

۲. ۵۷

۳. ۴۱

۴. ۸۱

پاسخ : ب

ابتدا تمامی یال ها را بر اسا وزنشان به صورت صعودی مرتب می کنیم سپس یال ها را به ترتیب انتخاب می کنیم به صورتی که حلقه ایجاد نکند

$$1 + 3 + 4 + 9 + 17 + 23 = 57$$

۱۴- کدام یک از الگوریتم‌های زیر حریصانه نیست؟

۱. Krushal

۲. Floyd

۳. Huffman

۴. Dijkstra

پاسخ: ب

روش حریصانه برای حل الگوریتم‌های زیر کاربرد دارد:

الگوریتم بقیه دادن پول  $\square$  الگوریتم prime  $\square$  الگوریتم کروسال  $\square$  الگوریتم دیکسترا  $\square$  الگوریتم کوله پشتی  $\square$   
الگوریتم زمان‌بندی  $\square$  الگوریتم هافمن  $\square$  الگوریتم‌های ادغام بهینه

۱۶-  $G=(V,E)$  یک گراف بدون جهت و همبند و وزن دار است و  $a$  و  $b$  دو رأس مجزا در آن فرض کنید  $p_1$  مسئله‌ی پیدا کردن کوتاه‌ترین مسیر بین  $a$  و  $b$  و  $p_2$  مسئله‌ای پیدا کردن بلندترین مسیر ساده بین  $a$  و  $b$  باشد، کدام

یک از گزینه‌های زیر در مورد  $p_1$  و  $p_2$  درست است؟

۱.  $p_1$  و  $p_2$  را می‌توان در زمان چند جمله‌ای حل کرد

۲.  $p_1$  و  $p_2$  را نمی‌توان در زمان چند جمله‌ای حل کرد

۳.  $p_1$  را می‌توان در زمان چند جمله‌ای حل کرد اما  $p_2$  را نمی‌توان حل کرد.

۴.  $p_2$  را می‌توان در زمان چند جمله‌ای حل کرد اما  $p_1$  را نمی‌توان حل کرد.

پاسخ: ج

کوتاه‌ترین مسیر بین دو رأس را می‌توان توسط الگوریتم فلوید یا دیکسترا در زمان چند جمله‌ای محاسبه کرد ولی بلندترین مسیر را در زمان چند جمله‌ای نمی‌توان حل کرد.

۱۸- کدام یک از الگوریتم‌های زیر برای حل مسائل بهینه‌سازی به کار می‌رود؟

۱. پویا -حریصانه

۲. تقسیم - حل

۳. تقسیم و حل -حریصانه

۴. پویا - تقسیم و حل

پاسخ: الف

در اغلب الگوریتم‌های پویا مسئله بهینه سازی موضوعی کلیدی است. الگوریتم‌های حریصانه هم اغلب برای مسائل بهینه سازی کاربرد دارند.

۲۰- روش ..... برای حل مسائلی استفاده می‌شود که در آن‌ها یک دنباله از اشیاء از یک مجموعه مشخص انتخاب می‌شود، به طوری که دنباله ملاکی را در بر می‌گیرد؟

۱. شاخه و قید

۲. عقبگرد

۳. تقسیم و حل

۴. حریصانه

پاسخ : ب

اغلب مسائلی که روش عقبگرد حل می‌شوند، از نوعی هستند که از اصول مفاهیم، نمایش، پیمایش و جستجو درخت‌ها سود می‌برند، سود می‌برند، بیشتر مسائلی که توسط گراف و درخت حل می‌شوند، مسائلی از نوع تصمیم‌گیری هستند. پس بیشتر مسائلی که توسط روش بازگشت به عقب حل می‌شوند، از نوع مسائل تصمیم‌گیری هستند. روش عقبگرد برای حل مسائلی استفاده می‌شود که در آن‌ها یک دنباله از اشیاء از یک مجموعه مشخص انتخاب می‌شود به طوری که این دنباله ملاکی را در بر می‌گیرد.

۲۲- راهبرد عقبگرد، کدام مسئله را بسیار بهبود می‌بخشد و بسیار مناسب است؟

۱. ضرب ماتریس‌ها

۲. کوله پشتی

۳. دورها میل‌تونی

۴. کوله پشتی صفر و یک

پاسخ : د

روش عقبگرد برای حل مسائل زیر کاربرد داد:

مسئله n وزیر ✕ مسئله حال جمع زیر مجموعه‌ها ✕ مسئله رنگ‌آمیزی گراف‌ها ✕ مسئله مدارهای میل‌تونی ✕ مسئله کوله پشتی صفر و یک

۲۴- الگوریتم شاخه و قید را در نظر بگیرید کدام گزینه صحیح است؟

۱. تکامل یافته‌ای از روش حریصانه است.

۲. تکامل یافته‌ای از روش پویا است.

۳. بهترین روش برای حل مسائل حریصانه است.

۴. تکامل یافته‌ای از روش عقبگرد است.

روش انشعاب و تحدید بهبود یافته‌ی روش عقبگرد می‌باشد

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

تشریحی

۲- از الگوریتم عقبگرد برای مسئله رنگ آمیزی گراف، برای یافتن همه رنگ آمیزی‌ها ممکن گراف زیر با سه رنگ قرمز، سبز و آبی استفاده نمایید. عملیات را مرحله به مرحله نشان دهید.

در رنگ آمیزی گراف می‌بایست رئوس طوری رنگ شود که هیچ دو رأس مجاوری هم‌رنگ نباشد

۴- الف- روش کار الگوریتم ادغام (merge) را برای ادغام دو آرایه مرتب شرح دهید.

ب- پیچیدگی زمانی مرتب سازی ادغامی (merge sort) را در بدترین حالت تحلیل نمایید.

الف) تابع merge این تابع ۲ آرایه را به عنوان ورودی می‌گیرد سپس این دو آرایه را به صورت مرتب شده در آرایه سوم ادغام می‌کند شمارنده آرایه اول i و شمارنده آرایه دوم j و شمارنده آرایه سوم k می‌باشد بدین صورت که عنصر i آرایه ۲ مقایسه می‌کند هر کدام کوچکتر بودند در محل k آرایه سوم قرار می‌دهد سپس شمارنده آن عنصر و نیز شمارنده k یک واحد افزوده می‌شود و به همین ترتیب ادامه می‌یابد

(ب)

$$T(n) = \begin{cases} \theta(1) & n = 1 \\ 2T(\frac{n}{2}) + \theta(n) & n > 1 \end{cases}$$

که در آن  $T(\frac{n}{2})$  زمان بازگشت و حل بوده و  $\theta(n)$  زمان لازم برای ادغام می‌باشد.

$$T(n) = \begin{cases} a \\ 2T(\frac{n}{2}) + Cn \end{cases} \rightarrow \begin{cases} a = 2 \\ b = 2 \rightarrow \begin{cases} a & b^k \\ 2 = 2^1 \end{cases} \\ k = 1 \end{cases} \rightarrow \theta(n^k \log n)$$

$$k = 1 \rightarrow \theta(n \log n)$$