بسم الله الرحمن الرحیم

نام و نام خانوادگی : سجاد حیدری

شماره دانشجویی: 949761914

نام استاد : سید علی رضوی ابراهیمی

نام درس : تحلیل و طراحی الگویتم

موضوع : سوالات فرد تابستان ۹۵ و زوج نیمسال دوم ۹۶-۹۷

سوالات تستی

1) مرتبه زمان اجرا الگوریتم زیر کدام است ؟

```
(i<100000){
 for I =1;
  while(j=1;j<I;j++)
      j*=2;
   i*=3
}
```

1. O(1) 2. O(n) 3. $O(\log_3 n * \log_2 n)$ 4. $O(n^2)$

جواب: مرتبه زمانی اجرا الگوریتم O(1) است.

2) کدام گزینه رابطه بازگشتی مسئله برج هانوی را نشان میدهد؟

1. $T(n) = 2^n-1 , T(n) = 2T(n-1)+1$
2. $T(n) = \log n , T(n) = 2T()+1$
3. $T(n) = 2^{n-1} , T(n) = 2T(n-1)+1$
4. $T(n) = n + \log n , T(n) = 2T()+1$

جواب: در مسئله برج هانوی در هر بار اجرا ،دو بار n-1 دیسک و یک بار،یک دیسک جابجا می‌شود لذا برج هانوی : $T(n) = 2^n-1 , T(n) = 2T(n-1)+1$ می‌باشد، اگر رابطه ای بصورت $T(n) = aT(n-k)+b$ باشد، مرتبه زمانی آن برابر با $o(a^{n/k})$ خواهد بود پس جواب گزینه 1 است.

3) زمان متوسط در الگوریتم جستجوی خطی کدام است؟

1. $\frac{n}{2}$ 2. $\frac{n+1}{2}$ 3. n 4. log n +1

جواب: زمان متوسط در الگوریتم جستجوی خطی برابر است با $\frac{n+1}{2}$

4) پیچیدگی زمانی تابع زیر کدام است؟

$T(n) = T(\sqrt{n}) + 1$

1. $\theta(\log \log n)$
2. $\theta(\log \log \log n)$
3. $\theta(\log n)$
4. $\theta(n \log n)$

جواب: $T(n) = T(\sqrt{n}) + 1 \xrightarrow{n=2^m} T(2^m) = T\left(2^{\frac{m}{2}}\right) + 1 \Rightarrow T(n) = T\left(\frac{m}{2}\right) + 1$ با توجه به قیضه اصلی : a=1,b=2,k=0

$b^k = 2^0 \Rightarrow b^k = a$

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

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

5) تابع بازگشتی زیر بر روی درخت دودویی T چه کاری انجام می‌دهد؟

```
Int f (Node*T){
  If (T == NULL) return0;
  If (T → left == NULL && T → right==NULL) return 1;
else
   return 1 + f(T → left) + f(T → right);
}
```

1. شمارش تعداد گره‌های برگ درخت
2. شمارش تعداد گره‌های دو فرزندی درخت
3. شمارش تعداد گره‌های غیر برگ درخت
4. شمارش تعداد گره‌های درخت

جواب: تابع بازگشتی رو برو بر ری درخت دودوئی T کار شمارش تعداد گره‌هایی که در درخت است را انجام میدهد.

6) جواب رابطه باز گشتی کدام یا از گزینه‌های زیر است؟

$T(n) = 9T\left(\frac{n}{3}\right) + n$

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

جواب: از قضیه اصلی استفاده می‌کنیم :

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

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

a با b^k مقایسه می‌شود، و اگر $a<b^k$ باشد مرتبه زمانی $n^{\log_b a}$ خواهد بود، و اگر $a>b^k$ باشد مرتبه زمانی n^k خواهد بود، و اگر $a=b^k$ باشد مرتبه زمانی $n^k\log n$ خواهد بود.

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

7) مرتبه زمانی رابطه بازگشتی زیر کدام است؟

$$T(n) = \frac{n}{3} + d$$

1. $\theta(1)$ 2. $\theta(n)$ 3. $\theta(\log n)$ 4. $\theta(n \log n)$

جواب: مرتبه زمانی بازگشتی $T(n) = \frac{n}{3} + d$ برابر است با $\theta(\log n)$

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

1. $o(nlogn)$ 2. $o(n)$ 3. $o(n^2)$ 4. $o(\log n)$

جواب: نتایج زیر را برای الگوریتم مرتب سازی سریع داریم:

زمانی که داده‌های از قبل مرتب شده باشند، الگوریتم در بدترین حالت خود می‌باشد.

بدترین حالت	حالت میانگین	بهترین حالت	
$\theta(n^2)$	$\theta(nlogn)$	$\theta(nlogn)$	مرتب سازی

9) چه تعداد مقایسه برای جستجو عدد 84 به روش دودویی در آرایه زیر بنا است؟

11	12	18	20	21	23	27	40	75	80	85

1. 4 2. 5 3. 3 4. 2

جواب: 4 مقایسه برای جستجوی دودوئی عدد 84 به روش دودوئی در آرایه فوق موجود است.

10) بدتری حالت الگوریتم‌های تقسیم و حل، برای n ورودی کدام گزینه است؟

1. مسئله به تعدادی زیر مسئله تقسیم می‌شود.
2. مسئله به سه قسمت تقسیم شود.
3. مسئله به قسمت‌های مساوی تقسیم شود.
4. مسئله به n قسمت تقسیم شود.

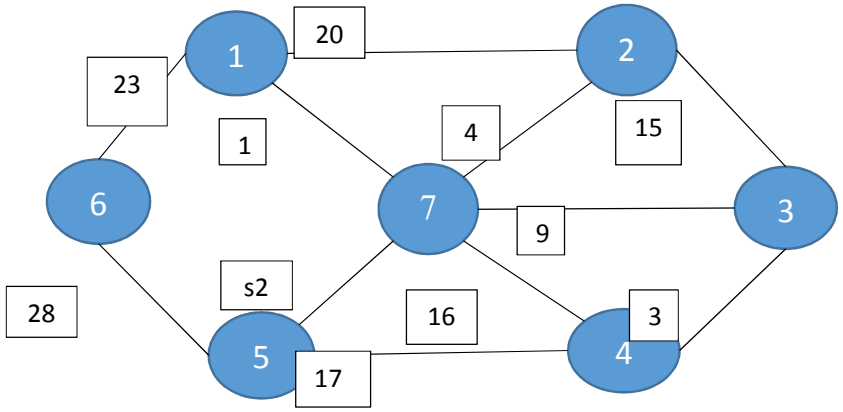
جواب: برای مسائلی که با تقسیم مسئله اصلی، مسائل کوچکتر دوباره به اندازه تقریبا n باشد و مسائلی که به تعداد زیادی زیر مسئله با طول n/c تقسیم می‌شوند، روش تقسیم و حل مناسب نیست.

11) حل مساله‌ای به اندازه n به کمک روش تقسیم و حل، هر مساله بزرگ به 3 زیر مساله به اندازه n-1 تقسیم می‌شود.الگوریتم شکستن و ادغام دارای زمان $o(1)$ است. مرتبه بزرگی این الگوریتم کدام است؟

1. $O(2^n)$ **2.** $O(n^2)$ **3.** $O(n^3)$ **4.** $O(3^n)$

جواب: در حل مساله‌ای که به اندازه n تا با استفاده از روش تقسیم و حل، که هر مساله بزرگ را به 3 زیر مساله که اندازه هر یک آنها به n−1 تقسیم می‌شود. الگوریتم شکستن و ادغام دارای زمان o(1) است. مرتبه بزرگی این الگوریتم برابر با $O(3^n)$ است.

12) هزینه درخت پوشای مینیمم گراف زیر چیست؟



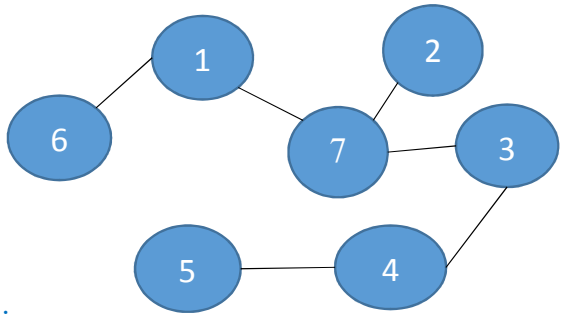
1. 68 **2.** 57 **3.** 41 **4.** 8

جواب: ابتدا تمامی یال‌ها را بر اساس وزنشان به صورت صعودی مرتب می‌کنیم سپس یال‌ها را به ترتیب انتخاب می‌کنیم به صورتی که حلقه ایجاد نکندُ مواردی که وزنشان با مشکی نوسته شده قابل قبول است

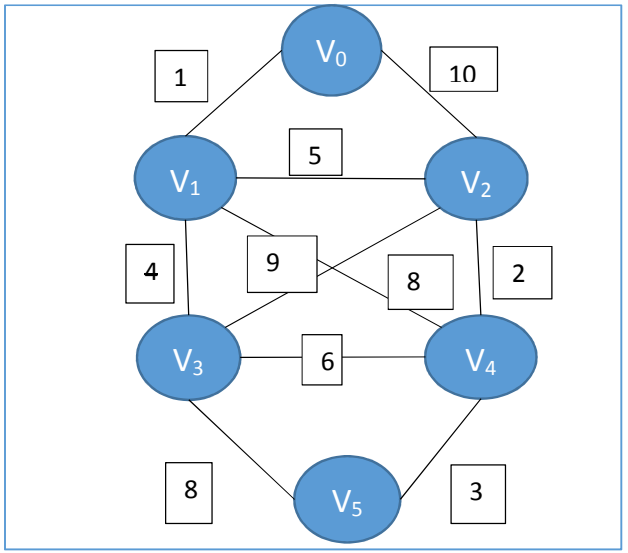
$(1,7)(3,4)(2,7)(3,7)(2,3)(4,7)(4,5)(1,2)(1,6)(5,7)(5,6)$

1 3 4 9 15 16 17 20 23 25 28

هزینه درخت پوشای مینیمم گراف 57 = 23 + 17 + 9 + 4 + 3 + 1



13) گراف زیر را در نظر بگیرید. برای یافتن درخت پوشای مینیمم این گراف به کمک الگوریتم پریم، کدام یال در مرحله سوم انتخاب می‌شود؟(شروع از v_0)



1. V_2V_4 **2.** V_1V_2 **3.** V_3V_4 **4.** V_4V_5

جواب: برای اینکه درخت پوشای مینیمم گراف فوق را با استفاده از الگوریتم پریم پیدا کنیم باید در مرحله سوم یال V_1V_2 را انتخاب کنیم.

14) کدام یک از الگوریتم‌های زیر حریصانه نیست؟

1. kruskal **2.** Floyad **3.** Huffman **4.** Dijkstra

جواب: روش حریصانه برای الگوریتم‌های زیر کاربرد دارد:

1- الگوریتم بقیه دادن پول 2- الگوریتم دیکسترا 3- الگوریتم هافمن 4- الگوریتم پریم 5- الگوریتم کوله پشتی 6- الگوریتم ادغام بهینه 7- الگوریتم کروسکال 8- الگوریتم زمانبندی

15) زمان لازم برای حل مساله زمانبندی با مهلت به روش حریصانه در بدترین حالت چقدر است؟

1. $\theta(2^n)$ 2. $\theta(n)$ 3. $\theta(n \log n)$ 4. $\theta(n^2)$

جواب: اگر بخواهیم به زمان لازم در حل مسئله زمانبندی با مهلت به روش حریصانه را برای بدترین حالت بیان کنیم باید بنویسیم $\theta(n^2)$

16) $G=(V,E)$ یک گراف بدون جهت و همبند و وزن دار است و a و b دو راس مجزا در آن فرض کنید p_1 مسئله‌ی پیدا کردن کوتاه‌ترین مسیر بین a و b و pz مسئله‌ای پیدا کردن بلندترین مسیر ساده بین a و b باشد، کدام گزینه‌های زیر در مورد p_1 و p_2 درست است؟

1. p_1 و p_2 را می‌توان در زمان چند جمله‌ای حل کرد.
2. p_1 و p_2 را نمی‌توان در زمان چند جمله‌ای حل کرد.
3. p_1 را می‌توان در زمان چند جمله‌ای حل کرد اما p_2 نمی‌توان حل کرد.
4. p_2 را می‌توان در زمان چند جمله‌ای حل کرد اما p_1 نمی‌توان حل کرد.

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

17) کدام گزینه صحیح است؟

1. هر مساله بهینه سازی را می‌توان از روش برنامه نویسی پویا حل کرد.
2. در روش برنامه نویسی پویا، برای حل مسائل L تنها از مسائل سطح L-1 استفاده می‌شود.
3. در روش برنامه نویسی پویا، مسائل از بالاترین سطح به پایین ترین سطح حل می‌شود.
4. معمولا حل مسائل به روش تقسیم و حل بازدهی کمتری نسبت به روش برنامه نویسی پویا دارد.

18) کدام یک از الگوریتم‌های زیر برای حل مسائل بهینه سازی به کار می‌رود؟

1. پویا-حریصانه 2. تقسیم و حل 3. تقسیم و حل-حریصانه 4. پویا- تقسیم و حل

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

19) اگر الگوریتم فلوید برای یافتن کوتاه‌ترین مسیر های گرافی با ماتریس مجاورت زیر بکار برود. $D^{(2)}[5][3]$ کدام است؟

$$W = \begin{bmatrix} 1 & 2 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$$

1. 4 2. 6 3. 7 4. ∞

جواب: اگر از الگوریتم فلوراید برای یافتن کوتاه‌ترین مسیر هایی که در گراف با ماتریس مجاورت فوق استفاده می‌شود.برای $D^{(2)}[5][3]$ برابر با 7 است.

20) روش ..... برای حل مسائلی استفاده می‌شود که در آن‌ها یک دنباله از اشیاء از یک مجموعه مشخص انتخاب می‌شود،بطوریکه این دنباله ملاکی را در بر می‌گیرد؟

1. شاخه و قید 2. عقبگرد 3. تقسیم و حل 4. حریصانه

جواب: اغلب مسائلی که به روش عقبگرد حل می‌شوند،از نوعی اند که از اصول مفاهیم، نمایش، پیمایش و جستجوی درخت‌ها سود میبرند، روش عقبگرد برای حل مسائلی استفاده می‌شود که در آن‌ها یک دنباله از اشیاء از یک مجموعه مشخص انتخاب می‌شود،بطوریکه این دنباله ملاکی را در بر می‌گیرد.

21) در مساله حاصل جمع زیر مجموعه‌ها با 4 عدد صحیح ،تعداد گره‌ها در درخت فضای حالت چقدر است؟

1. 33 2. 9 3. 31 4. 7

جواب: اگر بخواهیم در مسئله‌ای که حاصل جمع زیر مجموعه ها با 4 عدد صحیح،تعداد گره‌هایی که در درخت فضای حالت این مسئله خواهیم داشت برابر با 31 است.

22) راهبرد عقبگرد،کدام مسئله را بسیار بهبود می‌بخشد و بسیار مفید است؟

1. ضرب ماتریس‌ها 2 . کوله پشتی 3. دور هامیلتونی 4. کوله پشتی صفر و یک

جواب: روش عقبگرد برای حل مسائل زیر کاربرد دارد:

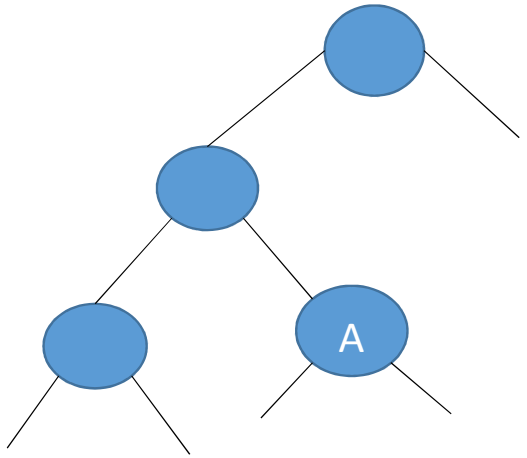
1- مسئله n وزیر 2- مسئله رنگ آمیزی گراف‌ها 3- مسئله کوله پشتی صفر و یک 4- مسئله حاصل جمع زیر مجموعه‌ها 5- مسئله هامیلتونی

عقبگرد مسئله کوله پشتی صفر و یک را بهبود می‌بخشد.

23) در مسئله کوله پشتی صفر و یک، فرض کنید $n = 5$، $w = 40$ و داشته باشیم :

i	1	2	3	4	5
p_i	8	5	15	10	20
w_i	16	15	25	8	15

اگر از تکنیک عقیگرد برای حل مساله استفاده شود، مقدار profit و bound از گره A از درخت فضای حالت چقدر است؟



1. Profit = 30 Bound = 40.2
2. Profit = 20 Bound = 35
3. Profit = 10 Bound = 38.5
4. Profit = 20 Bound = 18.4

جواب: برای مسئله فوق با مفروضات داده شده اگر بخواهیم از تکنیک عقبگرد استفاده کنیم مقدار Profit = 20 و Bound = 35 از گره A ادر درخت فضای حالت است.

24) الگوریتم شاخه و قید را در نظر بگیرید،کدام گزینه صحیح است؟

1. تکامل یافته‌ای از روش حریصانه است.
2. تکامل یافته‌ای از روش پویا است.
3. بهترین روش برای حل مسائل حریصانه است.
4. تکامل یافته‌ای از روش عقبگرد است.

جواب: روش انشعاب و تحدید بهبود یافته‌ی روش روش عقبگرد است.

25) کدام یک از مسائل زیر در کلاس p قرار می‌گیرد؟

1. فروشنده دوره گرد 2. رنگ آمیزی گراف 3. کوله پشتی صفر و یک 4. مرتب سازی آرایه ها

جواب: از مسائل فوق مرتب سازی آرایه‌ها در کلاس p قرار می‌گیرد.

سوالات تشریحی

1) آرایه زیر را به روش مرتب سازی سریع بصورت صعودی مرتب نمایید.(عملیات را مرحله به مرحله نشان دهید)

17	20	10	25	11	8	18	23

جواب: الگوریتم مرتب‌سازی در حالت کلی بصورت زیر روی لیست S با عناصر ارائه شده در بالا عمل می‌کند: عنصر محور کهولین عنصر است را انتخاب کن.

17 20 10 25 11 8 18 23

↑

عنصر محور

سپس: 1. عناصر کوچکتر از عنصر محور را در سمت چپ و عناصر بزگتر را در سمت راست لیست قرار بده.

8 10 11 17 20 25 18 23

↓ ↑ ↓

همه کوچکتر از عنصر محور | عنصر محور | همه بزرگتر از عنصر محور

2. زیر لیست‌های حاصل را با الگوریتم مرتب‌سازی سریع مرتب کن. 8 10 11 17 18 20 23 25

↑

عنصر محور

3. لیست حاصل مربت شدخه بصورت غیر نزولی است.

در بالاترین سطح بصور ذیل فراخوانی می‌شود($n = 8$): $QuickSort(0,7)$

مرحله اول: تابع تقسیم بندی با مقادیر زیر فراخوانی می‌شود: $Partitin(0,7,0)$

خروجی تابع partition : 8 10 11 17 18 20 18 25

↑

عنصر محور

و مقدار pivotpoint برابر 3 که به عنوان خروجی ارسال می‌شود، در این مرحله تابع بصورت $QuickSort$ بصورت زیر فراخوانی می‌شود $QuickSort(0,2)$

مرحله دوم: تابع تقسیم بندی با مقادیر زیر فراخوانی می‌شود: $Partitin(0,7,0)$

خروجی تابع partition :

8 10 11

↑

عنصر محور

و مقدار pivotpoint برابر صفر که به عنوان خروجی ارسال می‌شود، در این مرحله تابع بصورت $QuickSort$ بصورت زیر فراخوانی می‌شود. $QuickSort(0,-1)$

مرحله سوم: تابع $QuickSort$ با مقادیر زیر فراخوانی می‌شود: $QuickSort(1,2)$

در اینصورت تابع تقسیم بندی با مقادیر زیر فراخوانی می‌شود: $Partitin(1,2,0)$

خروجی تابع partition :

10 11

↑

عنصر محور

و مقدار pivotpoint برابر 1 بوده و به عنوان خروجی ارسال می‌شود، در این مرحله تابع بصورت $QuickSort$ بصورت زیر فراخوانی می‌شود $QuickSort(1,0)$

مرحله چهارم: تابع $QuickSort$ با مقادیر زیر فراخوانی می‌شود: $QuickSort(2,2)$

مرحله پنجم: تابع $QuickSort$ با مقادیر زیر فراخوانی می‌شود:(مقادیر لیست از اول تا عنصر محور تا حالا مرتب شده‌اند) $QuickSort(4,7)$

در اینصورت تابع تقسیم بندی با مقادیر زیر فراخوانی خواهند شد: $Partitin(4,7,0)$

خروجی تابع partition :

18 20 25 23

↑

عنصر محور

همچنین مقدار pivotpoint را برابر 5 قرار داده و به تابع $QuickSort$ ارسال می‌شود، در این مرحله تابع بصورت $QuickSort$ بصورت زیر فراخوانی می‌شود:

$QuickSort(0,-1)$

مرحله پنجم: تابع $QuickSort$ با مقادیر زیر فراخوانی می‌شود: $QuickSort(6,7)$

در اینصورت تابع تقسیم بندی با مقادیر زیر فراخوانی خواهند شد: $Partitin(6,7,0)$

خروجی تابع partition :

23 25

↑

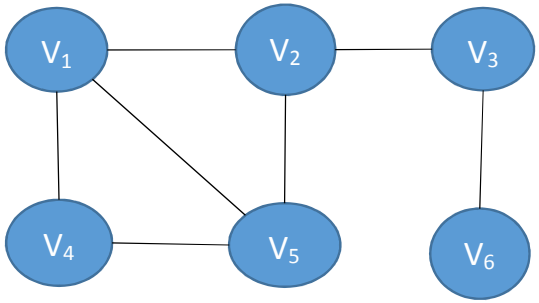
عنصر محور

و مقدار pivotpoint را برابر 7 بوده و به عنوان خروجی ارسال می‌شود، بنابراین کل خروجی بعد از انجام عملیات،بصورت ذیل خواهد بود:

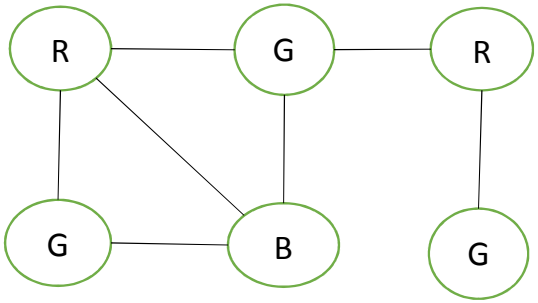
8 10 11 17 18 20 23 25

و این خروجی نهایی تابع $QuickSort$ با مقادیر 0 و 7 و لیست S با مقدیر بیان شده، خواهد بود؛ که یک لیست مرتب غیر نزولی است.

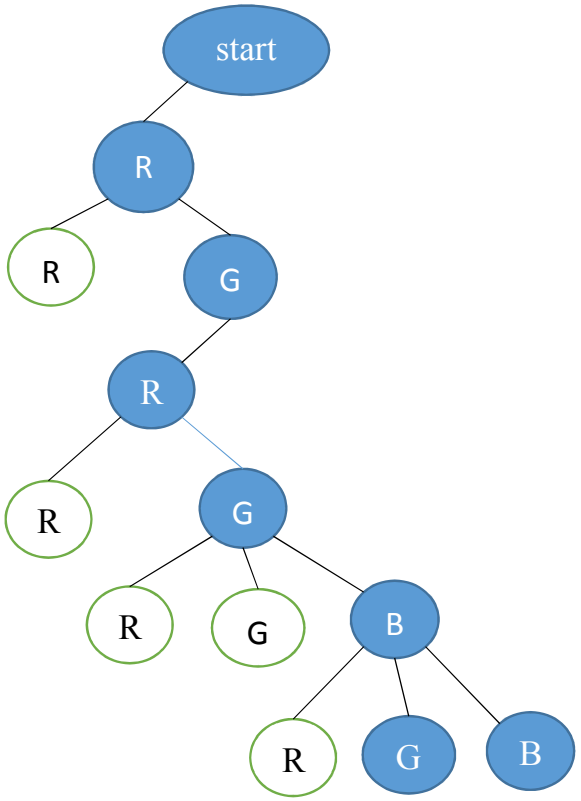
2) از الگوریتم عقبگرد برای مسئله رنگ آمیزی گراف، برای یافتن همه رنگ آمیزی‌های ممکن گراف زیر با سه رنگ قرمز،سبز و آبی استفاده کنید. عملیات را مرحله به مرحله نشان دهید.



جواب



جواب: در رنگ آمیزی گراف می‌بایست رئوس طوری طوری رنگ آمیزی می‌شود که هیچ دو راس مجاوری همرنگ نباشد.



3) با استفاده از روش برنامه نویسی پویا ضرب دو جمله ای $\binom{6}{4}$ را محاسبه نمایید.

جواب: ***ورودی:*** اعداد صحیح و مثبت 4 و 6 که در آن 6 => 4

خروجی: ضرب دو جمله‌ای $\binom{6}{4}$.

```
int     bin ( int 6 , int 4 )
{
```

```
    int i , j ;
    int B[6][4] ;
    for ( i = 0 ; i <= 6 ; i++)
        for ( j = 0 ; j <= min ( i , 4 ) ; j++ )
            if ( j == , ||j==i)
                B[i][j] = 1;
        else
                B[i][j] = B[i−1][j−1] + B[i−1][j] ;
     return B[6][4] ;
}
```

4) الف- روش کار الگوریتم ادغام را برای ادغام دو آرایه مرتب شرح دهید.

جواب: تابع merge : این تابع 2 آرایه را به عنوان ورودی می‌گیرد، سپس این 2 آرایه را به صورت مرتب شده در آرایه سوم ادغام میکند شمارنده آرایه اول i و شمارنده آرایه دوم j و شمارنده سومk می‌باشد،بدین صورت که عنصر i را با عنصر j آرایه 2 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 میابد.

A
6
5
3
1

B
5
4
2

=>

7
5
5
4
3
2
1

ب- پیچیدگی زمانی مرتب سازی ادغامی را در بدترین حالت تحلیل کنید.

جواب:

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

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

$$T(n) = \begin{cases} a \\ 2T\left(\frac{n}{2}\right) + cn \end{cases} \rightarrow \begin{cases} a = 2 \\ b = 2 \\ k = 1 \end{cases} \rightarrow \begin{cases} a \\ 2 \end{cases} = \begin{cases} b^k \\ 2^1 \end{cases} \rightarrow \theta(n^k logn)$$

$$k = 1 \rightarrow \theta(nlogn)$$

5) **از الگوریتم عقبگرد برای حل مسائل 4 وزیر استفاده کنید و عملیات را مرحله به مرحله نشان دهید. درخت فضای حالت هرس شده‌ای را نشان دهید که این الگوریتم تا نقطه رسیدن به حل نخست ایجاد کند.**

جواب: هدف از این مسئله چیدن n مهره (مثلا 4 وزیر) وزیر در یک صفحه شطرنج 4*4 است،بطوریکه هیچ دو وزیر یکدیگر را تهدید نکنند، به عبارت دیگر، هیچ دو مهره‌ای نباید در یک سطر، ستون و یا قطر یکسان نباشد

برای قرار دادن اولین وزیر در مجموعه 16 حالت وجود دارد، چنانچه اولین وزیر در یکی از مکان‌های خالی روی صفحه شطرنج قرار بگیرد برای دومین وزیر 15 حالت و سومین وزیر 14 حالت و چهارمین وزیر 13 حالت دارد،پس میتوان نوشت: $16! / 12! = 13 * 14 * 15 * 16$ به عبارت دیگر اگر درخت کلیه حالتها را رسم کنیم $16! / 12!$ برگ خواهد داشتو این همانظور که ملاحظه میکنید تعداد حالت بسیار زیادی می‌باشد و باید سعی کنیم تاحد ممکن از تولید بعضی از حالت‌ها جلو گیری کنیم،چون می‌خواهیم 4 وزیر بر روی یک صفحه شطرنج 4*4 قرار بگیرند، بطوریکه هیچ وزیری یکدیگر را تهدید نکنند. میتوان مسئله را به این صورت ساده کرد که هیچ دو وزیری نباید در یک ردیف قرار بگیرند. میتوان با نسبت دادن هر وزیری چک کردن اینکه کدام ترکیب از ستون ها به حل می‌انجامد،مسئله را حل کرد. چون هر وزیر را میتوان در یکی