

توجه:

- (۱) استفاده از هیچ منبعی در طول امتحان مجاز نمی باشد
- (۲) در صورتی که در سؤالی با ابهام روبرو شدید، فرض خود را بنویسید و متناسب با آن جواب را ارائه کنید.
- (۳) پس از اتمام (در مدت ۲ ساعت)، در اولین فرصت جواب‌های خود را در قسمت میان ترم ۱ در قسمت تمرین‌های درس بارگذاری کنید. (ذکر نام و نام خانوادگی فراموش نشود) پس از پایان وقت (از زمان شروع خودتان) مطلب جدیدی اضافه نشود.

در تمامی سؤالات ذکر بهترین نماد پیچیدگی ضرورت دارد

۱۰	<p>۱ پیچیدگی زمانی کدهای زیر را برحسب n محاسبه کنید. (نیازی به گزارش جزئیات بررسی نیست)</p> <p>در صورتی که کد دارای بهترین و بدترین زمان اجرا می باشد، گزارش هر دو و اینکه در چه حالتی این اتفاق صورت می‌پذیرد الزامی است. در صورتی که برای هر یک از موارد به مجموع تعدادی عناصر رسیدید که مرتبه ی آن را می دانید، نیاز به اثبات آن نیست و فقط کافیت مجموع را نوشته و نتیجه ی مجموع را گزارش کنید.</p>	۱				
	<table><tr><td><p>(الف)</p><pre>a = 0, b = n; while(a < b && x[a] == x[b]){ a++; b--; } sum = 0; for (i = a; i <= b; i++) for (j = 1; j <= b; j++) sum += x[i];</pre></td><td><pre>sum = 0; for (i = 1; i <= n; i++) for (j = 1; j <= i; j = j*2) sum++;</pre></td></tr><tr><td><p>(د)</p><pre>for (i=0; i<n; i++){ for (j=0; j<n; j++){ c[i][j] = 0; for (k=0; k<n; k++) c[i][j] += c[i][k]*c[k][j]; } }</pre></td><td><p>(ج)</p><pre>x = 0; for(i=1; i<=100 ; i++) for (j=1; j*j<=n ; j++) for (k=1; k<=n ; k = k*2) x++;</pre></td></tr></table>	<p>(الف)</p> <pre>a = 0, b = n; while(a < b && x[a] == x[b]){ a++; b--; } sum = 0; for (i = a; i <= b; i++) for (j = 1; j <= b; j++) sum += x[i];</pre>	<pre>sum = 0; for (i = 1; i <= n; i++) for (j = 1; j <= i; j = j*2) sum++;</pre>	<p>(د)</p> <pre>for (i=0; i<n; i++){ for (j=0; j<n; j++){ c[i][j] = 0; for (k=0; k<n; k++) c[i][j] += c[i][k]*c[k][j]; } }</pre>	<p>(ج)</p> <pre>x = 0; for(i=1; i<=100 ; i++) for (j=1; j*j<=n ; j++) for (k=1; k<=n ; k = k*2) x++;</pre>	
<p>(الف)</p> <pre>a = 0, b = n; while(a < b && x[a] == x[b]){ a++; b--; } sum = 0; for (i = a; i <= b; i++) for (j = 1; j <= b; j++) sum += x[i];</pre>	<pre>sum = 0; for (i = 1; i <= n; i++) for (j = 1; j <= i; j = j*2) sum++;</pre>					
<p>(د)</p> <pre>for (i=0; i<n; i++){ for (j=0; j<n; j++){ c[i][j] = 0; for (k=0; k<n; k++) c[i][j] += c[i][k]*c[k][j]; } }</pre>	<p>(ج)</p> <pre>x = 0; for(i=1; i<=100 ; i++) for (j=1; j*j<=n ; j++) for (k=1; k<=n ; k = k*2) x++;</pre>					
۸	<p>۲ درستی عبارات زیر را به کمک تعریف نمادهای پیچیدگی بررسی کنید. در صورتی که برای مقدار n_0 به تحلیل درستی رسیده باشید، محاسبه ی مقدار دقیق آن ضرورتی ندارد.</p> <p>1) $2^{n+2} = O(2^{n-2})$</p> <p>2) $2^n = \theta(2^{2n})$</p> <p>3) $n^2 - n = \Omega(n^2 + \log n)$</p> <p>3) $4^{\log_2 n} = \theta(5n^2)$</p>	۲				

۳	<p>پیچیدگی زمانی روابط زیر را با روش‌های گفته شده محاسبه نمایید. در همه ی موارد، پایه ی بازگشتی را می‌توانید به دلخواه مقداری ثابت در نظر بگیرید.</p> <p>مورد ۱ و ۲ با روشی دلخواه</p> <p>مورد ۳ با دوروش جایگذاری و تکرار - استقرا (فقط برای مرتبه ی $O(n^2)$)</p> <p>مورد ۴ و ۵ با روش قضیه ی اصلی (Master Theorem) و یا تعمیم آن</p> <p>1) $T(n) = 2T(\sqrt{n}) + O(\log(n)\log(\log(n)))$</p> <p>2) $T(n) = 3T(\frac{n}{2}) + 10\ln(n) + 20n^2$</p> <p>3) $T(n) = 4T(\frac{n}{2}) + n$</p> <p>4) $T(n) = \sqrt{2}T(\frac{n}{2}) + O(\lg(n))$</p> <p>5) $T(n) \leq 9T(\frac{n}{3}) + n^3$</p>
۴	<p>تابع زیر بر روی آرایه ی n عنصری A بصورت $\text{solve}(A, 0, n-1)$ فراخوانی شده است.</p> <p>الف) این تابع چه عملی را انجام می دهد؟</p> <p>ب) تابع زمانی $T(n)$ را برای این تابع بنویسید. نیازی به حل آن نیست. (در صورتی که بهترین و بدترین حالت داشته باشد، هر دو را با ذکر حالت بوجود آورنده ی آن بنویسید.)</p> <pre> int solve(int A[],int a,int b){ if (a==b){ if (A[a]>0) return 1; else return 0; } int m = (a+b)/2; int n1 = solve(A,a,mid) ; int n2 = solve(A,mid+1,b) ; int res = max(n1,n2) ; int n3 = 0,p; p = mid; while(p>=a && A[p]>0){ p--; n3++; } p = mid+1; while(p<=b && A[p]>0){ p++; n3++; } res = max(res,n3) ; return res; } </pre>
۴۰	موفق باشید - کشت کاران