

## ۱ سوال اول

رای هر گروه از توابع، توابع را به ترتیب افزایشی از پیچیدگی  $asymptotic(big - O)$  مرتب کنید.

۱.  $f_1(n) = n^{0.999999} \cdot \log(n)$  •

$f_2(n) = 100000000n$  •

$f_3(n) = 1.0000001^n$  •

$f_4(n) = n^2$  •

۲.  $f_1(n) = 2^{2^{1000000}}$  •

$f_2(n) = 2^{1000000n}$  •

$f_3(n) = \binom{2}{n}$  •

$f_4(n) = n\sqrt{n}$  •

## ۲ سوال دوم

می خواهیم در رشته ای از حروف کوچک انگلیسی، بلندترین زیر رشته ای که در آن هیچ حرفی دو بار تکرار نشده است را پیدا کنیم. شبه کدی بنویسید که این کار را در مرتبه  $O(n)$  انجام دهد. (برای راحتی می توانید فرض کنید که رشته به صورت آرایه ای از اعداد ۰ تا ۲۵ به شما داده شده است)

## ۳ سوال سوم

اگر ماتریس  $n * n$  از اعداد داشته باشیم، آیا الگوریتمی وجود دارد که بتواند  $peak$  را بیابد ( $peak$  در یک ماتریس مکانی با ویژگی است که چهار همسایه آن (شمال، جنوب، شرق و غرب) مقدار کمتر یا مساوی از

آن مکان دارند.)

۱. در  $O(n)$ ؟

۲. در  $O(n \log(n))$ ؟

۳. در  $O(n^2)$ ؟

در هر یک از موارد بالا در صورت وجود *pseudocode* الگوریتم را بنویسید در غیر این صورت بگویید چرا ممکن نیست.

## ۴ سوال چهارم

فرض کنید توابع  $f(n)$  و  $g(n)$  دو تابع غیر منفی باشند. با استفاده از نماد  $\theta$  در حالت پایه اثبات کنید که:

$$\max(f(n), g(n)) = \Theta(f(n) + g(n)) \bullet$$

## ۵ سوال پنجم

توضیح دهید که چرا عبارت "زمان اجرای الگوریتمی حداقل  $O(n^2)$  است." معنی ندارد.