

باسمه تعالی دانشگاه تهران - دانشکده مهند*سی* برق و کامپیوتر



ساختماندادهها و الگوريتمها

تمرین اول - پیچیدگی و الگوریتم های بازگشتی

حامد میرامیرخانی، ماردین نیچی تاریخ تحویل: ۱۴۰۲/۸/۵

۱.

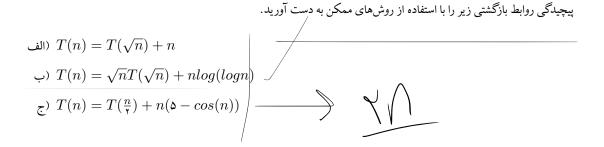
پیچیدگی زمانی قطعه کدهای زیر را محاسبه کنید.

٧.

روابط زير را رد يا اثبات كنيد.

$$f(n) = O(g(n)) \Rightarrow \mathbf{r}^{f(n)} = O(\mathbf{r}^{g(n)})$$
 (الف $f(n) \in O(g(n)) \Rightarrow g(n) \in \Omega(f(n))$ (ب $f(n) \neq O(g(n)) \Rightarrow f(n) = \Omega(g(n))$ (ح) $f(n) \neq O(g(n)) \Rightarrow f(n) = \Omega(g(n))$

۳.



۱۰ نمره

توابع زیر را براساس پیچیدگی زمانی مرتب کنید.

$$\log(n), n^n, \sum_{i=1}^{\frac{n}{\mathsf{T}}} n - \mathsf{Y}i, \log(n!), \mathsf{Y}^n, \sqrt[\mathsf{T}]{n}$$

$$(\log(\log(n)), n^{\log(\log(n))}, n^\pi, (\log(n))!$$

$$(\log(n)), n^{\log(\log(n))}, n^\pi, (\log(n))!$$

$$(\log(n), n^n, n^{\log(\log(n))}, n^\pi, (\log(n))$$

۵.

پیچیدگی زمانی توابع بازگشتی زیر را محاسبه کنید.

func(n) {
if n <= 1
 return
for i in 1 to n*n:
 //0(1)
func(n-2);</pre>

(الف



func(a, b):
if(b == 0):
 | return a
 return func(b, a % b)

٠٠ نمره

فرض کنید یک دنباله ی $a_1, a_2, ..., a_n$ از اعداد در اختیار دارید. الگوریتمی از مرتبه O(n) طراحی کنید که بتواند حاصل عبارت مقابل را محاسبه کند.

$$\sum_{1 \le l \le r \le n} f(l,r)$$

که f(l,r) را اینگونه تعریف می کنیم:

$$f(l,r) = \sum_{i=l}^{r} a_i$$