



تمرین شماره ۱
Time Complexity
and
Recursion



ساختمان های داده و الگوریتم - پاییز 1401

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراح تمرین : **علی کرامتی**

پایان روز 7 آبان

استاد: دکتر هشام فیلی

۱. پیچیدگی زمانی هر یک از قطعه کدهای زیر که در زبان cpp نوشته شده اند را از روش **سیگما** نویسی محاسبه

کنید. (راه حل شهودی به تنهایی قابل قبول نیست و تنها می تواند تکمیل کننده پاسخ شما باشد) (15 نمره)

الف)

```
if (n % 2 == 0) {  
    for (int i = 0; i < n; i++) {  
        for (int j = i; j > 0; j--) {  
            for (int k = n; k >= 1; k /= 2) {  
                count *= 3;  
            }  
        }  
    }  
}
```

ب)

```
for (int i = 1; i < n; i++) {  
    int j = 1;  
    while(j * j <= i){  
        j++;  
    }  
}
```

```
bool isSymmetric(int mat[][MAX], int N)
{
    for (int i = 0; i < N; i++){
        for (int j = 0; j < N; j++){
            if (mat[i][j] != mat[j][i])
                cout << "Matrix is not symmetric!" << endl;
            return true;
        }
    }
    return false;
}
```

(ج)

۲. در هر مورد توابع را برحسب درجه رشدشان مرتب کنید. (از روش حدگیری در بی نهایت تنها برای چک

کردن درستی پاسخ نهایی خود می توانید استفاده کنید و صرف نوشتن حد بدون راه حل، نمره ای به شما

تعلق نخواهد گرفت) (24 نمره)

(الف)

$$n^2, \left(\frac{5}{4}\right)^n, (\log n)^2, \log n!$$

(ب)

$$\log n!, \sqrt{2}^{(\log n)^2}, n^{\log \log n}, (\log n)!$$

(ج)

$$\sum_{j=1}^n \sum_{i=1}^j i, n^{\frac{1}{\log n}}, \sum_{k=0}^n \frac{n^k}{k!}, 2^n$$

۳. هر یک از موارد زیر را اثبات یا رد کنید. (24نمره)

الف) $f(n) \neq O(g(n)) \Rightarrow f(n) = \Omega(g(n))$

ب) $f(n) \in O(g(n)) \Rightarrow O(\log_2 f(n)) \in O(\log_2 g(n))$

ج) $f(n) = O(g(n)), g(n) = O(h(n)) \rightarrow f(n) = O(h(n))$

د) $\log_k n = \theta(\log_2 n)$

۴. پیچیدگی زمانی روابط بازگشتی زیر را با روش قضیه اصلی محاسبه کنید. (28نمره)

الف) $T(n) = 9T(\frac{n}{3}) + n^2 \log^3 n$

ب) $T(n) = 2^n T(\frac{n}{2}) + n^n$

ج) $T(n) = \sqrt{n} T(\sqrt{n}) + n$

د) $T(n) = T(\frac{n}{2}) + n(5 - \cos(n))$

۵) فرض کنید در مساله برج های هانوی، تنها بتوان حلقه ها را بین میله های مجاور حرکت داد. به عبارت دیگر،

حلقه ها نمی توانند مستقیماً بین میله های مبدا و مقصد جا به جا شوند. یک تابع بازگشتی برای این مسئله نوشته

و پیچیدگی زمانی تابع خود را تحلیل کنید. (10نمره)

۶) الگوریتم یا شبه کدی ارائه دهید که تقسیم بلند را در پایه ای ثابت انجام می دهد و باقی مانده و خارج

قسمت نهایی را برگرداند. (پیچیدگی زمانی الگوریتم شما باید در اردر خطی باشد).

برای مثال تقسیم بلند 721 بر 64 مشابه زیر است (اگر عادت دارید مقسوم علیه را در سمت راست بنویسید به

میل خود جای آن را عوض کنید).

$$\begin{array}{r}
 0 \ 1 \ 1 \\
 \hline
 64 \overline{) 7 \ 2 \ 1} \\
 \underline{- 0} \\
 7 \ 2 \\
 \underline{- 6 \ 4} \\
 8 \ 1 \\
 \underline{- 6 \ 4} \\
 1 \ 7
 \end{array}$$

ابتدا الگوریتم چک می کند که چند بار 64 عدد 7 را می شمارد، زیرا عدد 7 بیش مرتبه ترین رقم عدد 721 است.

از آنجا که $64 > 7$ ، پس جواب 0 است. سپس الگوریتم بیش ترین مرتبه رقم باقیمانده یعنی 2 را به سمت راست

باقیمانده تقسیم قبلی یعنی 7 اضافه میکند و مشابه کار قبلی را انجام می دهد. در اینجا 64 یک بار عدد 72 را

می شمارد و باقیمانده عدد 8 می شود. باقیمانده پایانی باقیمانده کل تقسیم است.

نکات تکمیلی

- پاسخ های خود را تا زمان معین شده در سایت آپلود نمایید.
- هدف این تمرین یادگیری شماسست. لطفا تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق با قوانین درس با آن برخورد خواهد شد.
- دقت فرمایید که پاسخ سوال ها یکتا نیست و به دیگر پاسخ های صحیح نیز نمره تعلق می گیرد.
- در صورت وجود ابهام در مورد سوالات می توانید از طریق ایمیل با من در ارتباط باشید.

شاد باشید.