

بسمه تعالی



تمرین اول

تحلیل پیچیدگی الگوریتم

استاد: دکتر مریم لطفی

دستیاران آموزشی

سید حسین حسینی

امیر فیض

محمد امین آقا کبیری

مهدی شمس

ایمان خلیل الرحمانی

اریسا احسانی

شادی شهامت نیا

سید محمد حسین هاشمی

دانشکده مهندسی کامپیوتر زمستان 1402

1. نشان دهید به ازای هر مقدار ثابت a و b به طوری که $b > 0$ رابطه زیر برقرار است:

$$(n + a)^b = \theta(n^b)$$

2. پیچیدگی قطعه کدهای زیر را در بدترین حالت به دست آورید .
(کد اول)

```
1  int count = 0;
2
3  for (int i = N; i > 0; i /= 2) {
4      for (int j = 0; j < i; j++) {
5          count += 1;
6      }
7  }
```

(کد دوم)

```
1  for (int i = 1; i <= N; i++) {
2      for (int j = i; j <= N; j += i) {
3          cout << "Hi";
4      }
5  }
```

3. دو الگوریتم با نام های Alg1 و Alg2 برای مسئله ای با اندازه ورودی n وجود دارد. Alg1 در مدت n^2 میکروثانیه

و Alg2 در $100n\log(n)$ میکروثانیه اجرا می شود. Alg1 را یک برنامه نویس می تواند در چهار ساعت پیاده

سازی کند و برای کامپایل کردن آن دو دقیقه زمان CPU احتیاج دارد. در مقابل، Alg2 به 15 ساعت وقت

جهت پیاده سازی و شش دقیقه زمان CPU نیاز است. اگر به هر برنامه نویس ساعتی 20 دلار پرداخت شود و

همچنین زمان استفاده از CPU دقیقه ای 20 دلار ارزش داشته باشد، مشخص کنید که مسئله ای با اندازه ورودی

500 را چندبار باید با Alg2 حل کرد تا هزینه آن قابل توجیه باشد؟

4. توابع زیر را بر اساس پیچیدگی زمانی مرتب کنید.

$n!$, $n^2 \log n$, $(\log n)^2$, $\log \log n$, $4^{\log n}$, n^{2^n} , 2^2 , $(\log n)^{\log n}$

5. کامپیوتری در واحد زمان مسئله ای به اندازه 16 را که الگوریتم آن از مرتبه $2^n * n$ است حل می کند، اگر

سرعت کامپیوتری 131072 برابر شود، این کامپیوتر همان مسئله را با چه اندازه ای در واحد زمان حل می

کند؟

6. به ازای هر کدام از زوج توابع $f(n)$ و $g(n)$ با توضیح مختصری نشان دهید که آیا توابع $f(n)$ از O و امگا و θ تابع $g(n)$ هست یا خیر؟ (فرض کنید که c یک عدد ثابت و بزرگتر از 1 هست)

$f(n)$	$g(n)$	O	امگا	θ
$2^{n/2}$	2^n			
$\log n!$	$\log n^n$			
2^n	2^{n-2}			
$n^2 * 2^n$	3^n			
$(\log n)^{\log n}$	$4^{\log n}$			
c^n	n^c			
n^c	$(\log n)!$			

7. پیچیدگی زمانی الگوریتم زیر را محاسبه کنید.

```

1  for (i = n - 1; i > 0; i--) {
2      x = a[i];
3      a[i] = a[0];
4      p = 0;
5
6      while (True) {
7          c = 2 * p + 1;
8
9          if (c >= i)
10             break;
11
12         if (c != i - 1 && a[c] < a[i + 1])
13             c++;
14
15         if (x > a[c])
16             break;
17
18         a[p] = a[c];
19
20         p = c;
21     }
22
23     a[p] = x
24 }
25
26

```

موفق باشید