دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

تکلیف اول درس طراحی الگوریتم‌ها

نیم‌سال تحصیلی: بهار ۱۴۰۲

مدرّس: دکتر محمّدرضا حیدرپور

دستیاران آموزشی: مصطفی دریس‌پور - مجید فرهادی - محمّدیاسین کرباسیان - محمّدرضا مزروعی - امیر منصوریان - امیرارسلان یاوری

۱ مرتب‌سازی توابع

توابع زیر را براساس پیچیدگی زمانی مرتب نمایید. (۱۵ نمره)

$$n!, 4^{\log n}, e^n, n2^n, \binom{100}{n}, \log n^{\log n}, 2^{2^n}, 2^{3^{510000}}, \sqrt{2}^{\log n}, n^3$$

$$\binom{100}{n} < 2^{3^{510000}} < \log n^{\log n} < \sqrt{2}^{\log n} < 4^{\log n} < n^3 < n2^n < e^n < n! < 2^{2^n}$$

۲ مقایسه توابع

به ازای هر زوج تابع $f(x)$ و $g(x)$ مشخص کنید که تابع $f(x)$ از O، o، ω، Ω و Θ تابع $g(x)$ هست یا خیر. c و k اعدادی ثابت و بزرگ‌تر از 1 هستند. (۴۰ نمره)

$f(x)$	$g(x)$	O	o	ω	Ω	Θ
n^k	c^n	✓	✓	×	×	×
2^n	$2^{\frac{n}{2}}$	×	×	✓	✓	×
$\log n!$	$\log n^n$	✓	×	×	✓	✓
2^n	2^{n-2}	✓	×	×	✓	✓
$n2^n$	3^n	✓	✓	×	×	×
$\log n$	$\log^2 n$	✓	✓	×	×	×
$\log n$	$\log n^2$	✓	×	×	✓	✓
$n\log^2 n$	$\frac{n^2}{\log n}$	✓	✓	×	×	×

۳ بررسی عضویت در مجموعه توابع

گزاره‌های زیر را اثبات یا رد کنید. (۲۵ نمره)

- $\max(f(n), g(n)) \in \Theta(f(n)+g(n))$
  $\forall n \geq 0 : \frac{1}{2}(f(n)+g(n)) \leq \max(f(n), g(n)) \leq f(n)+g(n) \implies \max(f(n), g(n)) \in \Theta(f(n)+g(n))$
- $c > 1, 1+c+c^2+...+c^n \in \Theta(c^n)$
  $\forall n \geq 0 : c^n \leq 1+c+c^2+...+c^n \leq (n+1)c^n \implies 1+c+c^2+...+c^n \in \Theta(c^n)$
- $\log n \in O(\sqrt[3]{n})$
  $\forall n \geq e^{-3W(-\frac{\log_{10} 2}{3})} : \log n \leq \sqrt[3]{n} \implies \log n \in O(\sqrt[3]{n})$ $\quad W$ is product logarithm function
- $f(n) \in O(s(n)), g(n) \in O(r(n)) \implies \frac{f(n)}{g(n)} \in O\left(\frac{s(n)}{r(n)}\right)$
  $Counterexample : f(n) = n^2, s(n) = n^3, g(n) = n, r(n) = n^3$

- $f(n) \in O(s(n)), g(n) \in O(r(n)) \implies f(n) - g(n) \in O(s(n) - r(n))$

  $Counterexample : f(n) = n^2, s(n) = n^3, g(n) = n, r(n) = n^3$

۴ تحلیل برنامه

برنامه زیر را از لحاظ پیچیدگی زمانی تحلیل کنید. (۲۰ نمره)

```
#include <stdio.h>
void f(int n, int m)
{
        long long sum = 0;
        for (int i = 2; i < n; i *= 3)
        {
                for (int j = 0; j < m; j += 2)
                {
                        for (int k = 0; k < j; k++)
                        {
                                sum += 1;
                        }
                }
        }
        printf("%d\n", sum);
}
int main()
{
        int a;
        scanf("%d", &a);
        for (int i = 0; i < a; i++)
        {
                f(1 << i, i);
        }
        return 0;
}
```

$$f(n,m) = \sum_{i=1}^{\log n} \sum_{j=1}^{\frac{m}{2}} \sum_{z=1}^{2j} 1 = \log n \sum_{j=1}^{\frac{m}{2}} 2j \in O((\log n)m^2)$$

$$f(a) = \sum_{i=1}^{a} f(2^i, i) = \sum_{i=1}^{a} i^3 \in O(a^4)$$

۵ بهینه‌سازی

اگر $f(n) \in \Theta(P)$ و $g(n) \in \Theta(\frac{n}{P})$ باشد، P را طوری تعیین کنید که $h(n) = f(n) + g(n)$ کم‌ترین پیچیدگی زمانی ممکن را داشته باشد. (۲۰ نمره مازاد)

$$h(n) = f(n) + g(n) \in \Theta(\max(f(n), g(n))$$
$$\min(\max(f(n), g(n))) = \min(\max(P + \frac{n}{P}))$$
$$P = \frac{n}{P} \implies P = \sqrt{n}$$