

مسئله چهارم (۱.۵ نمره)

فرض کنید که یک لیست از اعداد طبیعی همچون a_1, a_2, \dots, a_n در دست داریم. هدف ما پیدا کردن بیشترین مقدار ممکن است اگر عملگرهای مجاز، ضرب دو عضو متوالی آرایه، و سپس جمع همه اعداد به دست آمده باشد. هر عضو می تواند حداکثر در یک عضو دیگر ضرب شود. با بهره گیری از رویکرد برنامه ریزی پویا، یک الگوریتم خطی برای حل این مسئله ارائه نمایید. دو نمونه ورودی و خروجی در زیر داده شده است. شما باید رابطه بازگشتی و حالات پایه را معین کنید. اگر قرار باشد نمونه داده شده دوم (که جواب آن ۱۹ است) را حل کنیم، چه زیرمسائلی باید حل شوند؟ آنها را معین کنید.

ورودی: 1, 2, 3, 1

خروجی: 8 (که به صورت $1 + (2 \times 3) + 1$ به دست می آید).

ورودی: 2, 2, 1, 3, 2, 1, 2, 2, 1, 2

خروجی: 19 (که به صورت $2 + (2 \times 2) + 1 + (3 \times 2) + 1 + (2 \times 2) + 1$ به دست می آید).

مسئله پنجم (۱.۵ نمره)

فرض کنید که یک نمونه از مسئله کوله پشتی در اختیار شما قرار گرفته است. شما n آئتم که وزن هریک از آنها برابر با w_i و ارزش هرکدام از آنها برابر با v_i است در اختیار دارید. ظرفیت کوله پشتی نیز W است. تفاوت این مسئله با مسئله کوله پشتی معمولی آن است که شما مجاز به انتخاب مکرر هریک از آئتم ها هستید، با این شرط که حداقل K نوع آئتم مختلف انتخاب شوند. به بیان دیگر، تحت این شرط که حداقل K نوع مختلف از آئتم ها انتخاب شوند، شما مجاز هستید که از هریک از آئتم ها به تعداد دلخواه بردارید. (عدد K به عنوان ورودی در اختیار شما قرار می گیرد). یک الگوریتم مبتنی بر رویکرد DP معرفی کنید که بیشترین سود ممکن را برای یک کوله پشتی با ظرفیت W که باید دربردارنده حداقل K نوع مختلف از آئتم ها باشد مشخص کند. راهنمایی: از یک آرایه سه بعدی مانند M استفاده کنید که در آن، عدد ذخیره شده در درایه $M[i, w, k]$ برابر با مقدار بیشترین سودی باشد که می توان با بهره گیری از صرفاً i آئتم اول، و با ظرفیت w به دست آورد، تحت این شرط که حداقل k نوع متفاوت از آئتم ها مورد استفاده قرار گیرند. رابطه بازگشتی برای به دست آوردن مقدار نظیر درایه $M[i, w, k]$ ، به همراه ابعاد آرایه سه بعدی M ، درایه هدف، و حالات پایه باید همگی به شکل دقیق معین شوند.

بسمه تعالی

دانشگاه صنعتی اصفهان - دانشکده مهندسی برق و کامپیوتر

آزمون میانترم درس طراحی الگوریتم‌ها (۴۰۲۲) - اردیبهشت‌ماه ۱۴۰۳ - زمان: ۱۲۰ دقیقه

شماره دانشجویی:

نام و نام خانوادگی: میرزا حسن خاوری

مسئله اول (۲ نمره)
دو گزاره اول را با ذکر مثال نقض رد، و گزاره سوم را اثبات کنید. (شما حق استفاده از ابزار حد را دارید).
گزاره اول: اگر $\log_2(f(n)) \in \Theta(\log_2(g(n)))$ ، آنگاه $f(n) \in \Theta(g(n))$.
گزاره دوم: اگر $f(n) \notin \Omega(g(n))$ ، آنگاه قطعاً $f(n) \in O(g(n))$.
گزاره سوم:

$$(\log_2(n))^3 \in O(\sqrt[3]{n}).$$

مسئله دوم (۱.۵ نمره)
الف. مقدار متغیر r عاقبت چند خواهد شد؟ توضیح: اندیس‌های k و j نباید در جواب نهایی شما ظاهر شوند، اما وابسته بودن جواب نهایی به اندیس i اشکالی ندارد.

```
int r = 0;
for(int i = 1; i <= n; i++)
    for(int j = i+1; j <= n; j++)
        for(int k = i+j-1; k <= n; k++)
            r++;
```

ب. پیچیدگی زمانی $T(n)$ حلقه تودرتوی زیر را مشخص کنید:

```
for(int i = n; i >= 1; i /= 2)
    for(int j = i; j >= 1; j /= 2)
        <body of the innermost for loop> //Needs  $\Theta(1)$ 
```

تذکر: برای سادگی می‌توانید فرض کنید که n توانی از ۲ است؛ مثلاً $n = 2^k$ که $k \in \mathbb{N}$. $T(n)$ باید به شکل تابعی از k یا n مشخص شود.

مسئله سوم (۱.۵ نمره)
الف. فرض کنید که یک آرایه به طول n در اختیار داریم که k مؤلفه آغازین آن همگی یک، و بقیه همگی صفر هستند ($k \leq n$)؛ پس این آرایه به شکل $\underbrace{[1, 1, \dots, 1]}_{k \text{ تا}}, \underbrace{[0, 0, \dots, 0]}_{(n-k) \text{ تا}}$ است. با بهره‌گیری از راهبرد تقسیم

و غلبه، الگوریتمی از مرتبه $\log_2(n)$ طراحی کنید که حاصل جمع عناصر آرایه را برگرداند. جزئیات را قید کنید. رابطه بازگشتی‌ای که پیچیدگی الگوریتم فوق را توصیف می‌کند چیست؟

ب. الگوریتم مبتنی بر راهبرد تقسیم و غلبه برای ضرب اعداد صحیح بزرگ را در نظر بگیرید. هریک از دو عدد صحیح ورودی دارای n رقم هستند. در این الگوریتم، هریک از دو عدد صحیح ورودی به دو عدد صحیح که طول هریک تقریباً برابر با $\frac{n}{2}$ است می‌شکند. اگر تمهیدی برای کاستن از تعداد فراخوانی‌های بازگشتی صورت نگیرد، آنگاه پیچیدگی زمانی every-case الگوریتم حاصل، از رابطه $T(n) = 4T(\frac{n}{2}) + \Theta(n)$ پیروی می‌کند. تحت همین فرض (یعنی عدم بهره‌گیری از ترفندی که تعداد فراخوانی‌های بازگشتی را کاهش دهد)، اگر هریک از دو عدد صحیح ورودی به k عدد که طول هریک تقریباً $\frac{n}{k}$ است بشکند، آنگاه پیچیدگی زمانی every-case الگوریتم حاصل به شکل بازگشتی چه خواهد بود؟ توضیح دهید.