



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

تکلیف پنجم درس طراحی الگوریتم‌ها

نیم‌سال تحصیلی: بهار ۱۴۰۲

مدرس: دکتر محمدرضا حیدرپور

دستیاران آموزشی: مصطفی دریس‌پور - مجید فرهادی - محمدیاسین

کرباسیان - محمدرضا مزروعی - امیر منصوریان - امیرارسلان یآوری

۱ برنامه‌ریزی خطی نامحدود

نشان دهید برنامه‌ریزی خطی زیر نامحدود است. (۵ نمره)

$$\begin{aligned} \max(x_1 - x_2) \\ -2x_1 + x_2 &\leq -1 \\ -x_1 - 2x_2 &\leq -2 \\ x_1, x_2 &\geq 0 \end{aligned}$$

با حل دستگاه معادلات فوق خواهیم داشت:

$$x_1 \geq \frac{4}{5} \quad \frac{3}{5} \geq x_2 \geq 0$$

که با توجه به نامحدودبودن x_1 و محدودبودن x_2 ، مقدار $\max(x_1 - x_2)$ نامحدود خواهد بود.

۲ سودوکو

الگوریتمی با رویکرد Backtracking برای حل یک سودوکو $n^2 \times n^2$ (که در آن تعدادی از خانه‌های جدول مقداردهی شده‌است) ارائه دهید. (۱۵ نمره)

Input :

	1		4		5		
4		7			6	2	
8	2		6			7	4
			1		5		
5							3
		4		5			
9	6			3		4	5
3		5			8		1
	7			2		3	

Output :

6	1	9	7	4	2	3	5	8
4	5	7	8	3	1	6	9	2
8	2	3	6	9	5	1	7	4
2	3	6	4	1	9	5	8	7
5	9	1	2	7	8	4	6	3
7	8	4	3	5	6	2	1	9
9	6	2	1	8	3	7	4	5
3	4	5	9	6	7	8	2	1
1	7	8	5	2	4	9	3	6

```
#define n 3
#define N n*n
bool promising(int grid[N][N], int row, int col, int num)
{
    for (int x = 0; x <= N - 1; x++)
        if (grid[row][x] == num)
            return false;
    for (int x = 0; x <= N - 1; x++)
```

```
        if (grid[x][col] == num)
            return false;
    int startRow = row - row % n, startCol = col - col % n;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            if (grid[i + startRow][j + startCol] == num)
                return false;
    return true;
}

bool sudoku(int grid[N][N], int row, int col)
{
    if (row == N - 1 && col == N)
        return true;
    if (col == N)
    {
        row++;
        col = 0;
    }
    if (grid[row][col] > 0)
        return sudoku(grid, row, col + 1);
    for (int num = 1; num <= N; num++)
    {
        if (promising(grid, row, col, num))
        {
            grid[row][col] = num;
            if (sudoku(grid, row, col + 1))
                return true;
        }
        grid[row][col] = 0;
    }
    return false;
}
```

۳ کوتاه‌ترین مسیر

گرافی جهت‌دار و بدون وزن مفروض است. یک برنامه‌ریزی خطی برای یافتن کوتاه‌ترین مسیر بین دو رأس دلخواه این گراف ارائه دهید. (۲۰ نمره)

با تعریف متغیر $x_{u \rightarrow v}$ که نشانگر وجود یا عدم وجود یال $u \rightarrow v$ در مسیر بین دو رأس است، داریم:

$$\begin{aligned} \min \sum_{u \rightarrow v} x_{u \rightarrow v} \\ \sum_u x_{u \rightarrow s} &= 0 & s \text{ is the source.} \\ \sum_u x_{s \rightarrow u} &= 1 & s \text{ is the source.} \\ \sum_u x_{u \rightarrow d} &= 1 & d \text{ is the destination.} \\ \sum_u x_{d \rightarrow u} &= 0 & d \text{ is the destination.} \\ \sum_u x_{u \rightarrow v} &= \sum_w x_{v \rightarrow w} & \text{for all vertices } v \neq s, d \\ x_{u \rightarrow v} &\in \{0, 1\} & \text{for all edges } u \rightarrow v \end{aligned}$$

۴ کاهش

مسئله Vertex Cover را به مسئله Set Cover کاهش دهید. (۲۰ نمره)

مجموعه جهانی مسئله Set Cover را برابر مجموعه تمام یال‌های گراف مسئله Vertex Cover قرار داده و یال‌های متصل به هر رأس گراف مسئله Vertex Cover را به عنوان یک زیرمجموعه در مسئله Set Cover در نظر می‌گیریم. با حل مسئله Set Cover، کمینه تعداد زیرمجموعه‌ها که اجتماع آن‌ها برابر مجموعه جهانی است به دست می‌آید که معادل است با کمینه تعداد رئوس گراف مسئله Vertex Cover که حداقل یک رأس از رئوس دو سر تمام یال‌ها جزو این رئوس است.

۵ کوئرا

به دو سوال کوئرا پاسخ دهید. (۴۰ نمره)

۶ کاهش مازاد

مسئله‌های Clique و Independent Set را به یک‌دیگر کاهش دهید. (۲۰ نمره مازاد)
مسئله Clique: گرافی ساده و بدون جهت مفروض است. مطلوب است حداکثر تعداد رئوسی که دو به دو مجاور باشند.
مسئله Independent Set: گرافی ساده و بدون جهت مفروض است. مطلوب است حداکثر تعداد رئوسی که میان دو به دو آن‌ها یالی وجود نداشته باشد.

برای حل هر کدام از این دو مسئله برای یک گراف، کافی است دیگری را برای مکمل آن گراف حل کنیم. زیرا مجاورت دوجه‌دوی رئوس یک گراف مقابل استقلال دوجه‌دوی آن‌ها قرار دارد که این تقابل با مکمل‌گیری حاصل می‌شود.