# HW7A

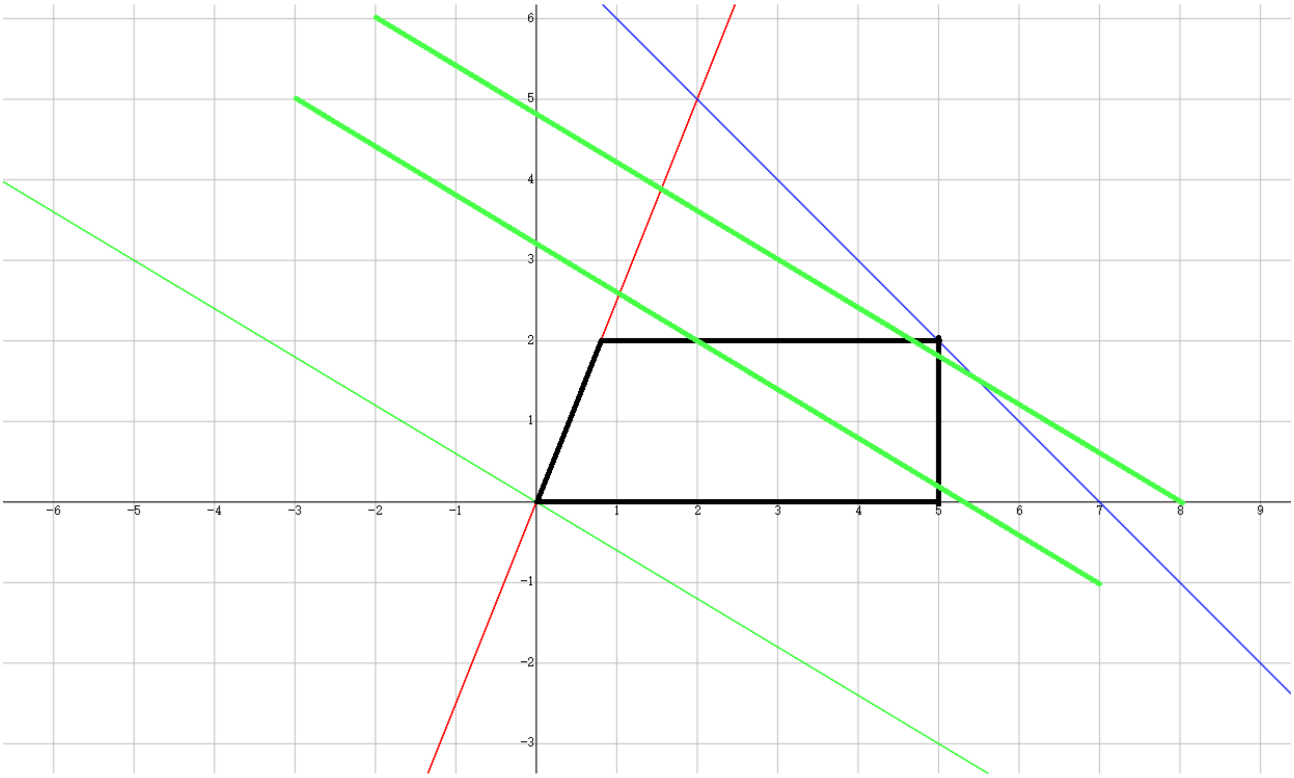
## Section II

Yanhao Wang 175002614

Zheng Qi 174003950

Mohan Xiao 174005865

## 7.1



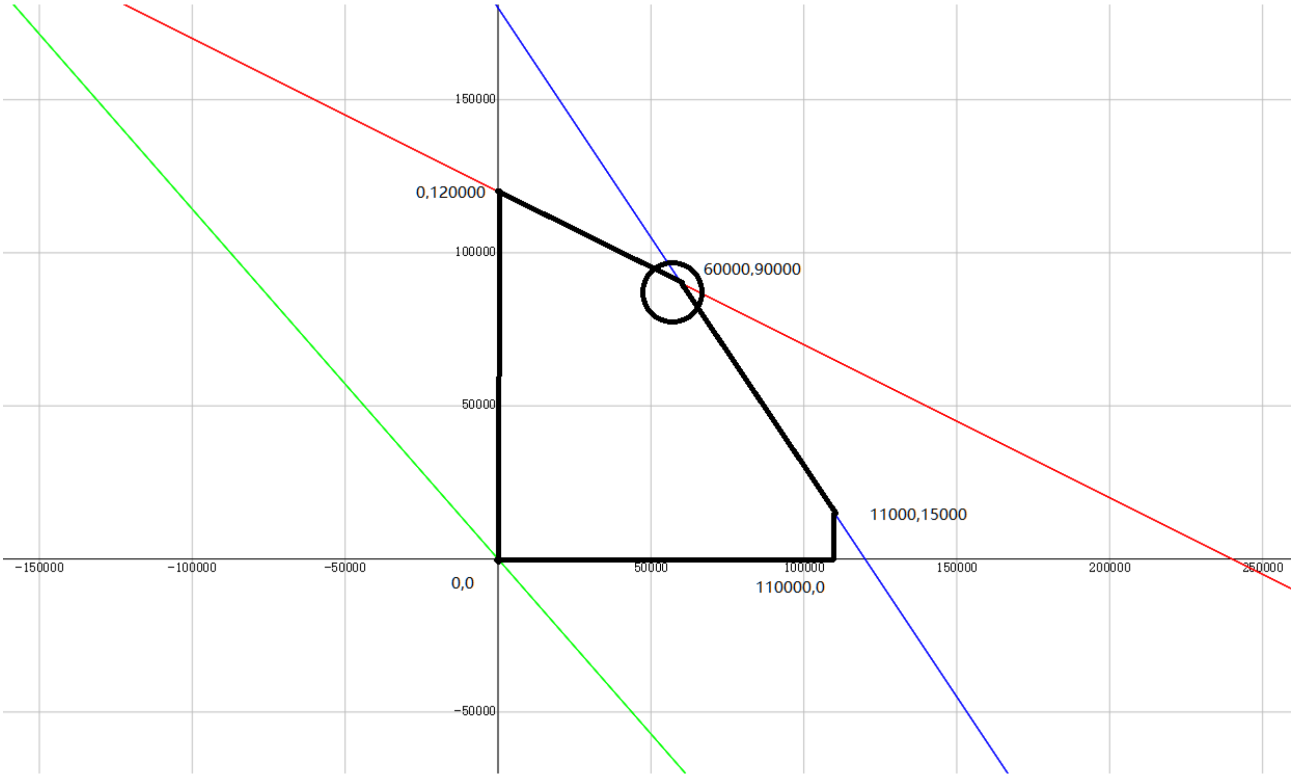

From picture we can see, the maximum value of the linear programming is when x=5,y=2, the result is 31.

## 7.5

a. x is the number of Frisky Pup, y is the number of Husky Hound.

$$
\begin{aligned}
&maximize && (7-1*1-2*1.5-1.4)x+(6-2*1-1*2-0.6)y \\
& && 1.6x+1.4y \\
&x+2y && \leq 240000 \\
&1.5x+y && \leq 180000 \\
&x && \leq 110000 \\
&x && \geq 0 \\
&y && \geq 0
\end{aligned}
$$

b.




The max profit is when x = 60000 y = 90000, profit is 222000

## 7.8

We can introduce a variable z to replace the formula. $z_m$ is the minimize

In standard form, the linear program is given by

Minimize $z_m$ for 1 to 7

$$\begin{aligned} z &\geq a*x_i + b*y_i - c \\ z &\geq -(a*x_i + b*y_i - c) \\ z_m &> z \end{aligned}$$

## 7.10

The maxumun flow is 13, the mathing cur is({$S, C, F$},{$A, B, D, E, G, T$})

## 7.12

$$\begin{aligned} &\max\ x_1 - 2x_3 \\ &\quad x_1 - x_2 \qquad \leq 1 \quad (a) \\ &\quad 2x_2 - x_3 \qquad \leq 1 \quad (b) \\ &\quad x_1, x_2, x_3 \qquad \geq 0 \quad (c) \end{aligned}$$

combine $(a)$, $(b)$, we can get:

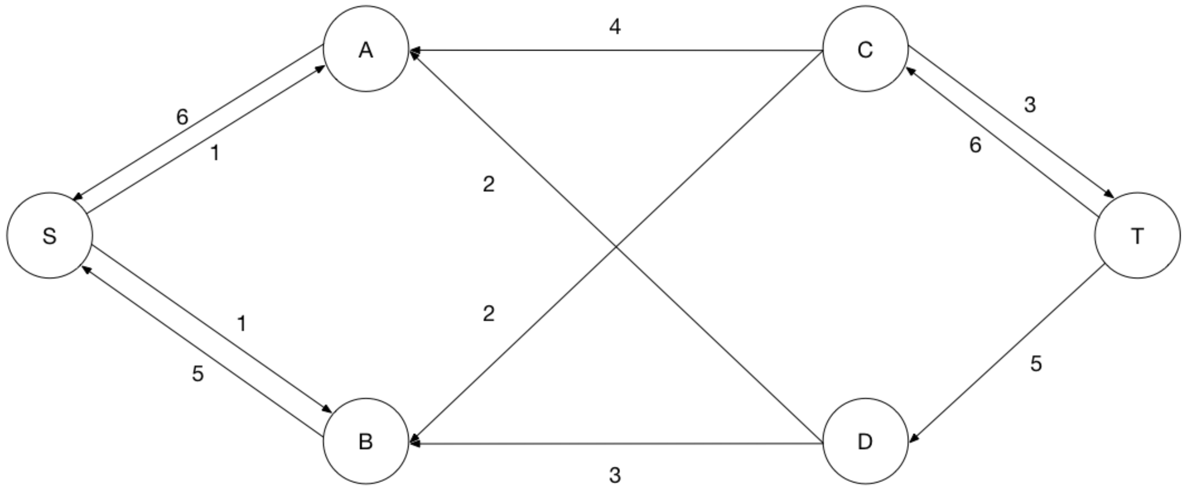
$$\begin{aligned} &2x_1 - x_3 && \leq 3 \\ &x_1 - x_3/2 && \leq 3/2 \\ &since\ x_3 \geq 0 \\ &x_1 - 2x_3 && \leq x_1 - x_3/2 \leq 3/2 \end{aligned}$$

Thus, the optimal solution for $x_1 - 2x_3$ is $3/2$, with $x_3 = 0$ when equality. So $(x_1, x_2, x_3) = (3/2, 1/2, 0)$ is optimal.

## 7.17

(a). Max Flow = 11, minimum cut = {$(S, A, B), (C, D, T)$}.

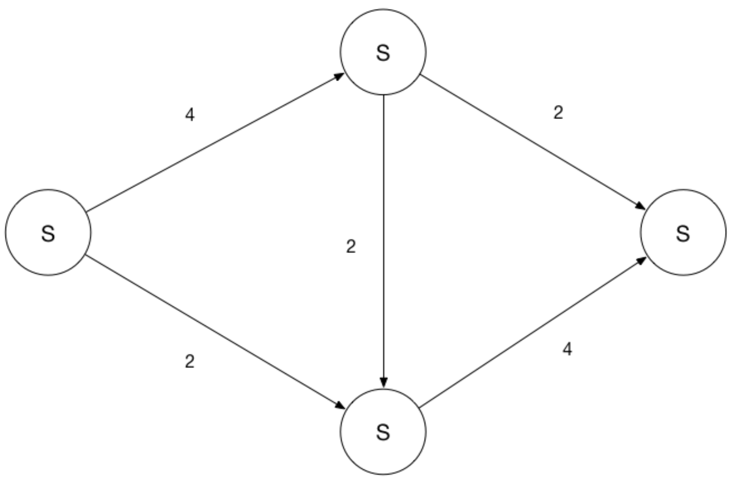
(b)




Nodes reachable from $S$ are {$A, B$}, nodes reachable from $T$ are {$S, A, B, C, D$}.

(c) Bottleneck edges are: $e(A, C)$ and $e(B, C)$. Inreasing both's capacity will increase the capacity going through node C, and thus increases the maximum flow.

(d)




(f)

1. Run the network flow algorithm and compute the residual graph R.
2. Run DFS on the residual graph R starting from node S. The nodes visited are those reachable by S, remember as set I.
3. Reverse the esidual graph R.
4. Run DFS from node T. Remember visited nodes as set J.
5. Last, go over all the vertices in the graph. If there exist an edge $e(u, v)$, which $u \in J, v \in I$, then that edge is a bottleneck edge.