Note 1

Q1 We can improve the ability of Student 1 to 3, and Student 3 to 2.

Q2 Input : Language ability $\ell_1, \dots, \ell_n$

Output : Improvement for each student $x_1, \dots, x_n$

Constraint : $x_i + x_j \geq 5 - \ell_i - \ell_j$ for all i, j

Objective Function : Minimize $\sum_i x_i$

Q3 Minimize $\sum_i x_p$

such that

$$x_1 + x_2 \geq 0$$
$$x_1 + x_3 \geq 2$$
$$x_2 + x_3 \geq 1$$

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Q4

$$A = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -3 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 20 \\ 30 \\ 40 \end{bmatrix} \quad c = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Input Matrix A, vectors b, c

Output : $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Constraint : $Ax \leq b$

Objective Function Maximize $c^T \cdot x$

Q5: $x_1 = 40$, $x_2 = 17.5$, $x_3 = 42.5$

Q6: $x_G = 6.6667$ $x_p = 0$

Q7: # of Gyudon should be integer.

Note 2

Q1 : Logic circuit with two layers ["or layer" and "and" layer]

Q2 : Assignment to each variable

Q3 : None

Q4 : Maximize # outputs from the "or layer" that are "true"

Q5 : $S1 = true$ $S2 = true$ $S3 = true$

Q6 : We know that Satisfiability is NP-Hard.

We can solve satisfiability using ~~our~~ an algorithm for our optimization model.

```
Assignment OurModel (Circuit C);
boolean Satisfiability (Circuit C) {
    Assignment A = OurModel (C);
    if output of C by Assignment A
        return true;
    else
        return false;
}
```

] Assignment "A" satisfies the maximum # or gate at the "or level". If there is any assignment that can satisfy all "or gate", we should have that as "A" from the library OurModel.

Q7 : If at least one of $\bar{S}_1$ or ~~$\bar{S}_2$~~ S_2 is true, we will have the output be true.

If $S_1 = 0$ or $S_2 = 1$, we will have the output true.

$\therefore$ If $(1 - S_1) = 1$ or $S_2 = 1$, we will have the output true.

Q8 : If all of the inequalities are satisfied, the output of all "or gate" will be true.

By that, the output of the "and gate" and the circuit will be true.

Q9 : We move 1 from the left side to the right side of the inequality.

Q9 (again) : ~~$x_1 = x_2 = x_3 = 0$~~ $S1 = S2 = S3 = 0$ $\Rightarrow$ $S1 = S2 = S3 = false$

This assignment satisfies the circuit.

Q10 ~~x_1, x_2, x_3 may be~~ The LP output may be something other than 0 and 1.

By that, we cannot converse $S1, S2, S3$ values to the assignment.

Note 3

Q1 Weight $w_1, \dots, w_n$, W , Happiness $h_1, \dots, h_n$, c

Q2 Ratio how much strawberry i is eaten $x_i \in [0, 1]$

Q3 $\sum_i w_i x_i \leq W$

Q4 Maximize $\sum_i h_i x_i - c \sum_{x_i > 0} h_i (1 - x_i)$

Q5

```
double[] Model(double[] w, double[] h, double W, double c)
Set knapsack(double[] w, double[] h, double W) {
    double[] x Model(w, h, W, ∞);
    return { i : x_i = 1 };
}
```

Q6 $w_1 = 2$ $W = 1$ $h_1 = 2$ $c = 0$

Q7 $x_1 = 0$ $x_2 = 1$ $\rightarrow$ happiness $= 2$

Q8 $x_1 = 1$ $x_2 = 0.0001$

Q9 $1.9999 - 20 \cdot 0.9999 \cdot 1.9999 \approx -38$

Q10 Because an optimal solution can give a negative objective value, 0.9-approximation algorithm may also give a negative one.

By that, there is no positive α such that $\underbrace{SOL}_{\text{can be negitive}} \geq \alpha \cdot OPT$.

Q11 Problem A does not have penalty, while our problem has.

Q12

$$OPT_{\text{Problem}} = \sum h_i \underbrace{x_i^{(p)}}_{\text{optimal solution of our problem.}} - c \sum_{x_i > 0} h_i (1 - x_i)$$

$$\leq \sum h_i x_i^{(p)}$$

$\because$ $x_i^{(k)}$ is an optimal solution of knapsack to minimize $\sum h_i x_i$

$$\leq \sum h_i \underbrace{x_i^{(k)}}_{\text{optimal solution of knapsack}}$$

$$= OPT_{\text{knapsack}}$$

Q13 There is no penalty. The objective function is same.

Q14 - 15 $SOL_{\text{Problem}} = SOL_B \geq 0.5\, OPT_A \geq 0.5\, OPT_{\text{Problem}}$

Q16 The approximation algorithm will choose items that have large $\frac{h_i}{w_i}$ before.

$$\underbrace{\frac{h_i^{(j)}}{w_i^{(j)}}}_{p_i} = \frac{h_i^{(j)}}{w_i^{(j')}} \le \frac{h_i^{(j')}}{w_i^{(j')}} \quad \text{when } j'<j$$

$p_i^{(j')}$ will be always added to the set before $p_i^{(j)}$.

Also, $p_i^{(j)}$ will never been selected at the last step as $h_i^{(j)} \le h_i^{(j')} \le \underbrace{\sum_{j'' \in S} h_{i''}^{(j'')}}_{\text{happiness in the set S before the last step.}}$

Q17 Optimal value of Bloom 3 is the largest value we can have.

Objective value of $\underbrace{S'}_{\text{a solution of Bloom3}}$ is just one of objective value.

Q18

$$\sum_{p_i^{(j)} \in S^*} h_i^{(j)} = \sum_i \sum_{p_i^{(j)} \in S^*} h_i^{(j)} = \sum_i \left[\underbrace{\sum_{p_i^{(j)} \in S^* : j \le 3} h_i^{(j)}}_{0.875} + \underbrace{\sum_{p_i^{(j)} \in S^* : j > 3} h_i^{(j)}}_{\le 0.125} \right]$$

If this term > 0

$$\le \frac{1}{0.875} \sum_i \sum_{p_j^{(j)} \in S'} h_i^{(j)}$$

$$= \frac{8}{7} \sum_{p_i^{(j)} \in S'} h_i^{(j)}$$

Q19

$$\underbrace{OPT_3}_{\text{optimal value of Bloom}_3} = \underbrace{\sum_{p_i^{(j)} \in S_3^*} h_i^{(j)}}_{\text{optimal solution of Bloom}_3} \ge \underbrace{\sum_{p_i^{(j)} \in S'} h_i^{(j)}}_{S^* \text{ cut short by argument in Q17–Q18}} \underset{\text{Q18}}{\ge} \frac{7}{8} \underbrace{\sum_{p_i^{(j)} \in S^*} h_i^{(j)}}_{\text{OPT of Bloom}_\infty} = \frac{7}{8} \underbrace{OPT_\infty}_{\text{optimal value of Bloom}_\infty}$$

$\boxed{\text{Solution of Bloom 3}}$

Q20

$$SOL_\infty \underset{\text{More items in } S^* \text{ than in } S'}{\ge} SOL_3 \ge 0.5\, OPT_3 \underset{Q_{19}}{\ge} \frac{7}{16} \cdot OPT_\infty$$

Q21 We need not to limit at 3. We can move further to consider $p_i^{(4)}$ if $p_i^{(3)}$ is taken to the set S. It is important to note that we do not need to consider all $p_i^{(j)}$ but $p_i^{(j+1)}$ when $p_i^{(j)}$ is taken. That makes the number of items to consider in each step equal N.