Q1

a)

min $\Sigma_{i=1}^{m} w_i x_i$

subject to

$\forall a \in$ U, $\Sigma_k x_k \geq 3$ for $a \in S_k$, $1 \leq$ k $\leq$ m

$x_i \in \{0, 1\}$ for $1 \leq i \leq m$

$x_i \in \{0, 1\}$ indicates whether student i is selected to be a representative.

For each activity a, having $\Sigma_k x_k \geq 3$ for some k such that student k participates in activity a satisfies the constraint that for each activity in U, at least three of the four students involved in that activity must be selected. Minimizing $\Sigma_{i=1}^{m} w_i x_i$ satisfies the goal of minimizing the total workload of the students selected as representatives.

b)

LP relaxiation:

min $\Sigma_{i=1}^{m} w_i x_i$

subject to

$\forall a \in$ U, $\Sigma_k x_k \geq 3$ for $a \in S_k$, $1 \leq$ k $\leq$ m

$x_i \leq 1$ for $1 \leq i \leq m$

$x_i \geq 0$ for $1 \leq i \leq m$

For LP optimal solution x_i^*, let $\hat{x}_i = 1$ if $x_i^* \geq 0.5$, and $\hat{x}_i = 0$ otherwise.

Claim: Rounded solution is a feasible solution to the integer program.

Would like to show that for each activity $a \in$ U, suppose students h, i, j, k participates in activity a, at least three of $\{x_h^*, x_i^*, x_j^*, x_k^*\}$ are at least 0.5, so that at least three of $\{\hat{x}_h, \hat{x}_i, \hat{x}_j, \hat{x}_k\}$ are 1.

Proof: Suppose, on the contrary, it is possible that less than three of $\{x_h^*, x_i^*, x_j^*, x_k^*\}$ are at least 0.5, in other words, it is possible that two or more have values less than 0.5. Suppose, without loss of generality, x_h^* and x_i^* are less than 0.5. Then $x_h^* + x_i^* < 1$. Since $0 \leq x_j^* \leq 1$ and $0 \leq x_k^* \leq 1$, then $x_h^* + x_i^* + x_j^* + x_k^* < 3$, contradicts the constraint $x_h^* + x_i^* + x_j^* + x_k^* \geq 3$

Claim: Rounded solution provides 2-approximation.

Proof: Since some $x_i^* \in [0.5, 1]$ is shifted up to 1, then $w_i \hat{x}_i \leq 2 w_i x_i^*$, then $\Sigma_{i=1}^{m} w_i \hat{x}_i \leq 2\Sigma_{i=1}^{m} w_i x_i^*$. Since $2\Sigma_{i=1}^{m} w_i x_i^* = 2$ * LP optimal value ≤ 2 * ILP optimal value, then $\Sigma_{i=1}^{m} w_i \hat{x}_i \leq 2$ * ILP optimal value.

Q2

a)

Following is the counter-example: $m = n = 3, p_{1,1} = 1, p_{1,2} = 4, p_{1,3} = 1, p_{2,1} = 4, p_{2,2} = 5, p_{2,3} = 4, p_{3,1} = 1, p_{3,2} = 4, p_{3,3} = 1$

1 4 1
4 5 4.
1 4 1

The greedy algorithm would pick $p_{2,2} = 3$ and $p_{1,1}, p_{1,3}, p_{3,1}, p_{3,3} = 1$, resulting with a total profit 9. However the optimal solution is to pick $p_{1,2}, p_{2,1}, p_{2,3}, p_{3,2} = 2$, and the optimal profit is 16.

b)

Let S be the selection returned by the greedy algorithm and let T be an optimal solution. Define M to be an empty multiset. For each $(i, j) \in T$

Case 1: $(i, j) \in S$. Add (i, j) to M.

Case 2: $(i, j) \notin S$, then we know from the greedy algorithm (i, j) was removed from C. Thus, there exist $(i', j') \in S$, (i', j') is adjacent to (i, j), and $p_{i',j'} \geq p_{i,j}$. Add (i', j') to M.

Claim 1: Profit(M) $\geq$ Profit(T).
Proof: from how M is constructed, for each element e $\in$ T, e is either in M, or an adjacent corner e' is in M such that $p_{e'} \geq p_e$.

Claim 2: 4Profit(S) $\geq$ Profit(M)
Proof: From how M is constructed, for each element m $\in$ M, m $\in$ S.
For each element e' in S
Case 1: e' is also in T and thus only one instance of e' is in M
Case 2: e' is not in T but e' $\in$ S. From how M is constructed, e' is adjacent to some e $\in$ T. Since e' can only be adjacent to 4 different elements in T, therefore there are at most 4 instances of e' in M.

Combine claim 1 and claim 2 we have 4Profit(S) $\geq$ Profit(T). Hence, the greedy algorithm gives 4-approximation.

Q3

a)

A randomized 1/2-approximation algorithm for Exact Robust-Max-3-SAT: Set each variable to true with probability 0.5 and to false with probability 0.5

Proof for correctness:

For each clause C_i:

C_i is not satisfied if:

case 1: all three literals are false. Pr[case 1] $= \frac{1}{2^3} = \frac{1}{8}$

case 2: the first literal is false, the rest are true. Pr[case 2] $= \frac{1}{2^3} = \frac{1}{8}$

case 3: the second literal is false, the rest are true. Pr[case 3] $= \frac{1}{2^3} = \frac{1}{8}$

case 4: the third literal is false, the rest are true. Pr[case 4] $= \frac{1}{2^3} = \frac{1}{8}$

Pr[C_i is not satisfied] $= \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{1}{2}$

C_i is satisfied if: otherwise

Pr[C_i is satisfied] = 1 - $\frac{1}{2} = \frac{1}{2}$

Let τ denote the number of clauses satisfied of a random assignment that has m clauses.

E[τ] $= \Sigma_{i=1}^{m}$ Pr[C_i is satisfied] $= \Sigma_{i=1}^{m} \frac{1}{2} = \frac{m}{2}$

E[τ] $= \frac{1}{2} * m \geq \frac{1}{2} * OPT$, therefore this is a 1/2-approximation algorithm.

b)

Let τ denote the number of clauses satisfied of a random assignment that has m clauses. Suppose there are n variables.

DerandomizedAlgorithm:

```
for i = 1, 2, ..., n:
    expectationT = 0 // E[τ|x_1 = z_1, ..., x_{i-1} = z_{i-1}, x_i = T]
    for j = 1, 2, ..., m:
        if C_j already resolves to true:
            expectationT += 1
        else if there are 3 literals left in C_j:
            expectationT += 0.5
        else if there are 2 literals left in C_j and the third one is evaluated to be true:
            expectationT += 0.75
        else if there are 2 literals left in C_j and the third one is evaluated to be false:
            expectationT += 0.25
        else if there is 1 literal left in C_j: // the rest evaluated to be true & false each
            expectationT += 0.5
```

```
14      expectationF = 0 // E[τ|x_1 = z_1, ..., x_{i-1} = z_{i-1}, x_i = F]
15      for j = 1, 2, ..., m:
16          if C_j already resolves to true:
17              expectationF += 1
18          else if there are 3 literals left in C_j:
19              expectationF += 0.5
20          else if there are 2 literals left in C_j and the third one is evaluated true:
21              expectationF += 0.75
22          else if there are 2 literals left in C_j and the third one is evaluated to be false:
23              expectationF += 0.25
24          else if there is 1 literal left in C_j: // the rest evaluated to be true & false each
25              expectationF += 0.5
26      if expectationT ≥ expectationF:
27          z_i = T
28      else
29          z_i = F
30      x_i = z_i
```

Correctness:

As computed from part (a), after setting each variable to true with probability 0.5 and to false with probability 0.5, E$[\tau] = \frac{m}{2}$. Since x_1 is either assigned to be true or false, one of E$[\tau|x_1 = T]$ and E$[\tau|x_1 = F]$ is at least as good as E$[\tau]$. Then, depending on which leads to a better conditional exception (line 22 - line 26), assign x_1 to be either true or false. Once the right assignment is made for x_1, the same logic applies to x_2 and we make a choice for x_2... We repeat this process for each x_i $(1 \leq i \leq n)$, until each variable gets assigned a value. In the end, E$[\tau|x_1 = z_1, x_2 = z_2, ..., x_n = z_n]$ is at least as good as E$[\tau]$, which is $\frac{m}{2}$. Since after setting the values of all variables $x_1, ..., x_n$, τ becomes a constant, and the expectation of a constant is the value of the constant itself, then $\tau \geq \frac{m}{2}$. Therefore this DerandomizedAlgorithm always returns a truth assignment of variables satisfying at least $\frac{m}{2}$ clauses.

Worst case running time:

The for loop on line 1 executes n times, and inside this for loop body, there are two for loops on line 3 and 15 that each executes m times. The rest of the computation takes constant time, therefore the worst case running time is O(nm).