Exercise 1

a)In order to solve the problem we devised a greedy algorithm that tries to maximize the number of elves that have at least a friend that is a compliant officer. This is the algorithm:

```
V=[]
for e in W:
        V[e]=F_e
O=ø  ,  C=ø
for i in range(k):
        M=ø  ,  e_max=0
        for F_e in V:
                if |F_e \C|>|M|:
                        M=F_e  ,  e_max=e
        C=C ∪ M
        O=O ∪ e_max
return O
```

Santa knows for every elves $e \in W$ the set of his friends F_e so we firstly create a vector V that contains those F_e.Then we initialize to empty the set of compliant officers O and C ,that is used in the loops to store the elves that have at least a compliant officer friend. Then for k times (number of compliant officer elves to choose) we select the set F_e with the greatest difference in cardinality with C, so we select the set F_e with the highest number of elves without a compliant officer friend. We then add to C this selected F_e and we insert the corresponding elf e to O that in the end will contain the resulting set of k compliant officers.

b)

In order to prove that the proposed algorithm is a $1 - \frac{1}{e}$ approximation we firstly suppose that we know the optimal solution for k compliant officers that we denote as Opt.

We denote the difference between the optimum Opt and the selected sets C as R that at the beginning will be equal to Opt and once we select new F_e's will decrease .

So in every iteration R_i = Opt-C_i where with i we mean the i-th iteration to choose the i-th compliant officer.In every iteration we will add to C the new elements from the chosen F_e and this number of new elements in every iteration will be at least R_i/k. If there wasn't a set with at least R_i/k new elements to choose by the greedy algorithm in the i-th iteration then Opt wouldn't be the optimum so we would have a contradiction.

The new difference from the Opt in the following iteration will then be at most $R_i - \frac{R_i}{k}$.

So we can write the following: $R_{i+1} \geq R_i - \frac{R_i}{k} \rightarrow R_{i+1} \geq R_i(1 - \frac{1}{k})$.

If we start from R_{i+1}=k then we can derive the following series of disequalities:

$R_{i+1} \geq R_i\left(1-\frac{1}{k}\right)$ so we have $R_k \geq R_{k-1}\left(1-\frac{1}{k}\right) \geq R_{k-2}\left(1-\frac{1}{k}\right)^2 \geq \cdots \geq R_0\left(1-\frac{1}{k}\right)^k$

R_o as stated above is equal to Opt because R_o=Opt - C_o and C_o at the beginning is ø.

If we consider k=∞ then we have that $(1 - \frac{1}{k})^k$ is equal to 1/e.

so R_k>=Opt*$\frac{1}{e}$. R_k states how much we are away from the optimum at the last iteration.

so Opt - Opt*$\frac{1}{e}$ will be our solution C_k and can be rewritten as Opt(1- $\frac{1}{e}$).

Opt/C_k will be equal to 1- $\frac{1}{e}$ so our algorithm approximates Opt with this coefficient.

Exercise 2

a)

In order to solve the problem we construct the following integer linear programming model.

$\min z$

$$\begin{cases} P_{i1} + P_{i2} + P_{i3} = 1 \\ P_{ij} \in \{0, 1\} \\ z \geq P_{ij} \cdot a_{ij} \end{cases}$$

where a_{ij} is $a_{ij} = \sum_{e \in P_{ij}} 1 \mid e \in E$

Pij are integer variables with values 0 or 1. The P_{ij} value represents the probability that we choose the company j to transport the good i. Only one company can be chosen to transport a specific good i so Pi1+Pi2+Pi3=1.

aij is an integer that represents the sum of the obstructed segment of road in the path of the j company for the transportation of the good i.

z is an integer that is greater or equal than aij if we choose the company j to transport the good i and represents an upper bound to the aij chosen(if Pij is 1). The model, when optimized, will choose the best Pij's to minimize the value z, in order to minimize the maximum number of obstructed segments across all the company's paths chosen(Pij=1).

Now to optimize the model we relax it allowing Pij's to be values between 0 and 1.

$\min z$

$$\begin{cases} P_{i1} + P_{i2} + P_{i3} = 1 \\ P_{ij} \in [0, 1] \\ z \geq P_{ij} \cdot a_{ij} \end{cases}$$

where a_{ij} is $a_{ij} = \sum_{e \in P_{ij}} 1 \mid e \in E$

We optimize the relaxed model and for every good i we select the company corresponding to the max{P_{i1},P_{i2},P_{i3}} , in case all the 3 values are the same we randomly select a company for the good i since there are no differences between them.The value that has been chosen will be rounded up to 1 , the other ones will be rounded down instead to 0.

b)

In order to prove that this relaxation leads to a 3 approximation we can consider the worst case scenario where for the good i we have that $P_{i1}=P_{i2}=P_{i3}$ so we have $a_{i1}=a_{i2}=a_{i3}$.

In this case we have the same probabilities of choosing a company so there is no difference in which one we end up rounding up.

The optimal of the relaxed model in this case will be $z1=a_{ij}*P_{ij}$ and p_{ij} will be ⅓ so $z1=⅓ *a_{ij}$.

When we round up we get that $z2=a_{ij}*1$ so $z2=a_{ij}$.

The approximation is the z of the rounded solution / z of the relaxed problem.

So $z2/z1 = a_{ij}/(1/3*a_{ij}) =3$.

So the rounded solution in the worst case scenario will be a 3 approximation of the relaxed one.

In all the other cases where $P_{i1}=P_{i2}=P_{i3}$ is not satisfied then the quotient between z2 and z1 will be <=3.

Exercise 3

a)

To reason about the probability of the graph G_k to have the same min-cut set of the original graph G we consider the Karger's algorithm for the global min-cut seen in class.
Given the proof of the book (ref.) we know that the correct min-cut (A,B) will actually be returned by the algorithm if no edge of the global min cut set is contracted in any of the iterations 1,2,...n-2. This can be translated in $P[E_1 \cap E_2 \cdots \cap E_{n-2}]$(given as E_i the probability that an edge of the global min-cut set is not contracted in iteration i), and unwinding it we have :

$$\left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right) \quad \cdots \quad \left(\frac{2}{4}\right) \quad \left(\frac{1}{3}\right)$$

Then from there we can easily use the same formula , letting the algorithm continue until $\sqrt{n}$ nodes instead of 2 so we always select an edge different than F in every iteration until we get to $\sqrt{n}$ nodes after n-$\sqrt{n}$ $\left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{\sqrt{n}-1}{\sqrt{n}+1}\right)$contractions so we use:

We can use the factorial and their properties to derive the final probability of G_k having the same min-cut set of the original graph G (so the probability of not contracting an edge of the global min-cut in any of the iterations until $\sqrt{n}$ nodes) as follows:

$$\frac{\frac{(n-2)!}{(\sqrt{n}-2)!}}{\frac{n!}{\sqrt{n}!}} = \frac{\sqrt{n}!(n-2)!}{n!(\sqrt{n}-2)!} = \frac{\sqrt{n}(\sqrt{n}-1)}{n(n-1)} = \frac{n-\sqrt{n}}{n^2-n}$$

b)

In order to find the probability of finding a correct min-cut set we multiply the probability of arriving at $\sqrt{n}$ nodes with a correct min-cut set (as above),to the probability of arriving at 2 nodes starting from $\sqrt{n}$ nodes with a correct min-cut set ,trying this last part $\sqrt{n}$ times.
$\frac{2}{n(n-1)}$is the probability of outputting the correct min-cut set with the original algorithm, then since we start from $\sqrt{n}$ nodes we replace n with $\sqrt{n}$ obtaining $\frac{2}{n-\sqrt{n}}.(1-\frac{2}{n-\sqrt{n}})^{\sqrt{n}}$is the probability of finding a wrong min-cut set starting from $\sqrt{n}$ nodes for $\sqrt{n}$ times, then if we subtract from 1 this value,we obtain the probability of finding a correct min-cut set in at least a run starting from $\sqrt{n}$ nodes.In the end $(\frac{n-\sqrt{n}}{n^2-n})(1-(1-\frac{2}{n-\sqrt{n}})^{\sqrt{n}})$ will be the probability of outputting the correct global min-cut set. If we want to approximate it we can use the hint:

$$\left(\frac{n-\sqrt{n}}{n^2-n}\right)\left(1-\left(1-\frac{2}{n-\sqrt{n}}\right)^{\sqrt{n}}\right) \geq \left(\frac{n-\sqrt{n}}{n^2-n}\right)\left(1-\left(1-\frac{1}{n-\sqrt{n}}\right)^{\sqrt{n}}\right) \geq \left(\frac{n-\sqrt{n}}{n^2-n}\right)\left(1-\left(1-\frac{\sqrt{n}}{n-\sqrt{n}}\right)\right) = \frac{\sqrt{n}}{n(n-1)} = \frac{1}{\sqrt{n}(n-1)} \geq n^{-\frac{3}{2}}$$

c)

The original algorithm will have n-2 contractions so running it 2 times will produce 2n-4 contractions. The algorithm variation instead will have $n-\sqrt{n}$ contractions to get at $\sqrt{n}$ nodes, then will have $\sqrt{n}(\sqrt{n}-2)$contractions to repeat $\sqrt{n}$ times the contractions starting from $\sqrt{n}$ nodes to 2. So in the end we have $2n-3\sqrt{n}$ total contractions.

The error probability of the algorithm variation is $1-(\frac{n-\sqrt{n}}{n^2-n})(1-(1-\frac{2}{n-\sqrt{n}})^{\sqrt{n}})$. or if approximated as above is equal to $1\text{-}n^{-\frac{3}{2}}$. $1-(\frac{2}{n(n-1)})^2$ is the error probability of the original algorithm runned twice and if approximated is equal to $1\text{-}n^{-2}$. So in the end the algorithm variation will be better because the error probability is lower when n grows.

Exercise 4