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 $|F_e \setminus C|>|M|$:
                        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 the set 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. If we reach the optimum before selecting k compliant officers we return less than k elves. If we have to return exactly k elves and we reach the optimum before the end we could select the others at random.

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*, so the maximum number of elves with at least a compliant officer friend.

We denote the difference between the optimum *Opt* and the selected sets in 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 we will have $R_i = Opt-C_i$ where with i we denote the i-th iteration in which we 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: $R_{i+1} \leq R_i - \frac{R_i}{k}$ that rewritten is $R_{i+1} \leq R_i(1-\frac{1}{k})$.

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

$$R_{i+1} \leq R_i\left(1-\frac{1}{k}\right) \text{ so we have } R_k \leq R_{k-1}\left(1-\frac{1}{k}\right), R_k \leq R_{k-2}\left(1-\frac{1}{k}\right)^2, \ldots, R_k \leq R_0\left(1-\frac{1}{k}\right)^k$$

R_0 as stated above is equal to *Opt* because $R_0=Opt - C_0$ and C_0 at the beginning is ø.

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

So $R_k \leq Opt*\frac{1}{e}$. Practically R_k states how much we are away from the optimum at the last iteration.

So our solution C_k will be at least $Opt - Opt*\frac{1}{e}$ and it can be rewritten as $Opt(1-\frac{1}{e})$ so our algorithm approximates a solution with ratio no more than $(1-\frac{1}{e})$.

Exercise 2

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

$\min z$

$$\begin{cases} p_{i1}+p_{i2}+p_{i3}=1 & \forall i \in G \\ p_{ij} \in \{0,1\} & \forall i \in G, \forall j \in \{1,2,3\} \\ R_{eij} = 1 \text{ if } e \in P_{ij} \text{ else } 0 & \forall_e \in E, \forall_i \in G, \forall j \in \{1,2,3\} \\ S_e = \sum_{i,j} \ p_{ij} \cdot R_{e_{ij}} & \forall_e \in E \\ z \geq S_e & \forall_e \in E \end{cases}$$

p_{ij} 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 $p_{i1}+p_{i2}+p_{i3}=1$.

R_{eij} is an input of the problem that states if a segment $e \in E$ is included in a path P_{ij}, its value can be 0(not included) or *1* (included).

S_e represents the number of paths chosen that use the road segment *e*: if P_{ij} is chosen (p_{ij} is *1*) and if the segment *e* is included in the path P_{ij} (R_{eij}=1) then we will add *1* to this sum for every *i, j*.

z is a value that is greater or equal than S_e for every *e* and represents an upper bound to the maximum S_e value that will depend on the chosen paths. The model, when optimized, will choose the best p_{ij}'s to minimize the value *z*, in order to minimize the maximum number of paths that use the same road segment *e*.

Now we relax the model from an integer linear programming model to a linear programming model allowing p_{ij}'s to be values between *0* and *1*.

$\min z$

$$\begin{cases} p_{i1}+p_{i2}+p_{i3}=1 & \forall i \in G \\ p_{ij} \in [0,1] & \forall i \in G, \forall j \in \{1,2,3\} \\ R_{eij} = 1 \text{ if } e \in P_{ij} \text{ else } 0 & \forall_e \in E, \forall_i \in G, \forall j \in \{1,2,3\} \\ S_e = \sum_{i,j} \ p_{ij} \cdot R_{e_{ij}} & \forall_e \in E \\ z \geq S_e & \forall_e \in E \end{cases}$$

b)

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 and we cannot choose more than one company. The value that has been chosen will be rounded up to *1* , the other ones will be rounded down instead to 0. In order to prove that this rounding algorithm leads to a 3-approximation we can consider the worst case scenario where once the relaxed linear programming is optimized, the probabilities of selecting the roads that pass through the most used segment are ⅓ each and for every good *i* at most a path across all the companies's paths, to deliver the good *i,* pass through it. In this case we can obtain a certain max S_e that is the optimum value z of the model. Given a good *i* for which exists a path of a company that passes through the most used segment, since we have ⅓ of probability to choose each of those companies, we cannot select the *max*$\{p_{i1},p_{i2},p_{i3}\}$ but we have to choose it randomly. In case we end up, randomly, rounding up all the probabilities of choosing the paths that pass through the most used segment to 1 instead of the others that don't pass through it, we obtain the value *max* S_e for the rounded solution that will be 3*max S_e.

given the rounded $\overline{p_{ij}} \leq 3p_{ij}$ we obtain a $\overline{S_e} \leq 3S_e$ and given the case described above the max S_e will be 3 times the original. In all the other cases the rounded max S_e will always be ≤ than 3 times the original maximum S_e so in the worst case the rounding algorithm gives a 3-approximation.

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[1] and slides we know that the correct min-cut will 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* so with probability k/|E| for each iteration. Every node has to be connected to at least k edges so the number of edges |E| will be at least ½ k*n otherwise the global min cut would not be of size k. Given this, k/|E| will be ≤2/n.
So the probability of reaching the correct global min cut 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 :

$$\Pr[E_1 \cap E_2 \cdots \cap E_{n-2}] \geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{4}\right)\left(1-\frac{2}{3}\right) = \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{2}{4}\right)\left(\frac{1}{3}\right)$$

Then 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 those in the global min cut in every iteration until we get to $\sqrt{n}$ nodes after n-$\sqrt{n}$ contractions so we use: $\left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{\sqrt{n}-1}{\sqrt{n}+1}\right)$

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 set 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 $\frac{2}{n-\sqrt{n}}$. Then$(1-\frac{2}{n-\sqrt{n}})^{\sqrt{n}}$ 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. Executing the algorithm variation$(\frac{n-\sqrt{n}}{n^2-n})(1-(1-\frac{2}{n-\sqrt{n}})^{\sqrt{n}})$ will be the probability of outputting the correct global min-cut set. If we want to approximate it with the hint when n is big:

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

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 $\sqrt{n}-2$ contractions (from $\sqrt{n}$ nodes to 2) to repeat $\sqrt{n}$ times. 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 $1 - n^{-\frac{3}{2}}$. $1-(\frac{2}{n(n-1)})^2$ is the error probability of the original algorithm runned twice and if approximated is equal to $1 - n^{-2}$.
So in the end the algorithm variation will be better because the error probability is lower when n grows. The variation of the algorithm has a bigger number of contractions instead.

[1] Kleinberg, J., & Tardos, E. (2005). *Algorithm Design* (1st ed.). Pearson. pp. 716-718

Exercise 4

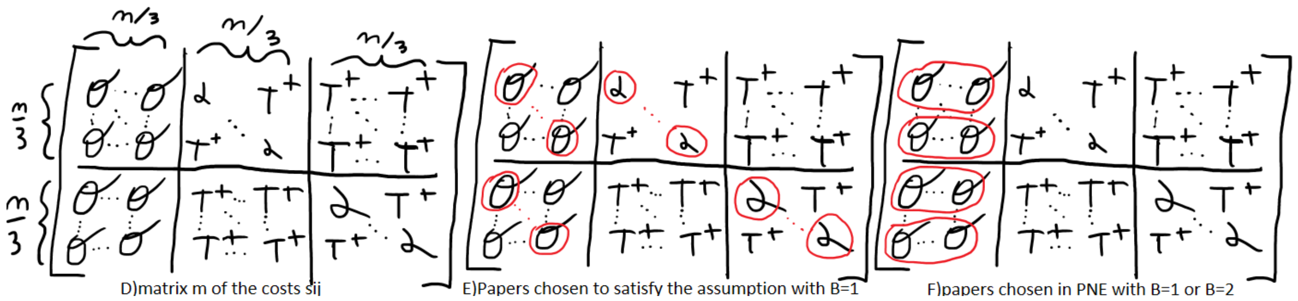
a)In order to prove that there are cases where the papers that get reviewed are close to 1/n we provide the following instance: *T = n, k = any, n = any*

$$\begin{pmatrix} \sigma & 1+\varepsilon & \cdots & 1+\varepsilon \\ \sigma & T+1 & \cdots & T+1 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma & T+1 & \cdots & T+1 \end{pmatrix} \begin{pmatrix} \boxed{\sigma} & 1+\varepsilon & \cdots & 1+\varepsilon \\ \boxed{\sigma} & T+1 & \cdots & T+1 \\ \boxed{\vdots} & \vdots & \ddots & \vdots \\ \boxed{\sigma} & T+1 & \cdots & T+1 \end{pmatrix} \begin{pmatrix} \boxed{\sigma} & \boxed{1+\varepsilon} & \boxed{\cdots} & \boxed{1+\varepsilon} \\ \boxed{\sigma} & T+1 & \cdots & T+1 \\ \boxed{\vdots} & \vdots & \ddots & \vdots \\ \boxed{\sigma} & T+1 & \cdots & T+1 \end{pmatrix}$$

A)matrix m of the costs sij's B)papers chosen in the PNE by the reviewers C)papers chosen to satisfy the assumption

In the matrix *m* above each row corresponds to a reviewer and each column corresponds to a paper. We fix the costs as above where in *m[i,j]* there is the cost s_{ij}. The papers with cost equal to *T+1* cannot be reviewed because are > *T*, so all the reviewers except the first one could review only the first paper. The cost of the first paper for every reviewer is σ, which is an infinitely small number. All the other papers for the first reviewer have a cost *1*+ε where ε is infinitely small. Given the assumption of the homework in which every reviewer has positive utility and all papers get at least one review, we can show that the reviewers can choose the papers (figure C) in order to satisfy that assumption. In this case, all the papers get at least a review and the utilities for each reviewer will be as follows: the first reviewer will have a utility equal to$(1/k - \sigma) - (\varepsilon \cdot n)$, then all the others will have $1/k - \sigma$(given both σ and ε greater than 0).$1/k - \sigma$ has to be >$\varepsilon \cdot n$: given 0< σ< 1/k we have to set $\varepsilon < (1/k - \sigma)/n$. In this way the first reviewer will have that the papers that he reviews (except the first one) will slightly decrease his total utility, but, given the constraints on σ and ε he will end up with an overall positive utility. In this way the assumption is verified but we are not in a PNE because the first reviewer, having seen all the other strategies, could play differently to get a higher utility. In the PNE (figure B) in fact, all the reviewers will choose the first paper and the first reviewer will not choose another paper because it would lower his utility. In the PNE the utilities are 1/k - σ that is bigger than the assumption utilities for the first reviewer (indifferent for the others) and seeing all the others strategies none of them would play differently to improve their utility so we are in a pure nash equilibrium with just a fraction 1/n of papers reviewed even though the assumption is respected as described before.

b)To solve the second point we created the following matrix that can be scaled for every n where in the columns there are the papers and in the rows the reviewers (n/3*n/3 costs in each block).



D)matrix m of the costs sij E)Papers chosen to satisfy the assumption with B=1 F)papers chosen in PNE with B=1 or B=2

We derive σ and α from the constraints and σ is infinitely small. k=2n/3,*T*=σ + α , σ**n/3*≤ *T* because we want a reviewer to be able to choose either all the papers with cost = σ or, one σ paper and a α paper to satisfy the assumption. We want 1<α<*(0.5 - σ+1)* to have positive utility when choosing a σ and a α paper for the assumption(0.5 is the reward for a paper σ when *b=1*). When *b=1* in the PNE the first *n/3* papers are reviewed because the α paper lowers the utility. We set:

$\left(\frac{2}{n\cdot\frac{2}{3}}-\sigma\right)\cdot\frac{n}{3} > \frac{2}{n\cdot\frac{2}{3}}-\sigma+2-\alpha,\ so\ ,\alpha > 1+\frac{3}{n}+\sigma\left(\frac{n}{3}-1\right)$ so when *B=2* choosing all the σ papers is better than choosing an α and a single σ paper, so even if we increase the reward, the reviewed papers in the PNE are 1/3.All the constraints together when choosing α and σ solve the problem.In other instances the papers reviewed improve(es point a if B=2),in this case they are still ⅓ (worst case).