Surnames	**Name**	**DNI**

AMMM Final Exam   Time: 3 hours   Jan. 8th, 2020

- *The exam contains 8 pages and 3 exercises*
- *Please fill in your name and DNI at the top of each exercise*
- *Write down the answers inside the rectangular boxes and tables*
- *Hand out all three exercises separately*

Exercise 1: Simplex Method **(2.5 pts)**

Consider the following linear optimization problem $P(\alpha)$ parameterized by α:

$$\begin{aligned} &\text{maximize } (3-2\alpha)x_1 + (\alpha-3)x_2 - x_3 \\ &\text{subject to} \\ &\qquad x_1 + 2x_2 - 3x_3 + x_4 = 5 \\ &\qquad 2x_1 + x_2 - 4x_3 + x_5 = 7 \\ &\qquad x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

1. **(1 pt)** Apply the simplex method to find an optimal solution for $P(3.5)$. Start at the following tableau

	x_1	x_2	x_3	x_4	x_5	
x_4	1.00	2.00	-3.00	1.00	0.00	5.00
x_5	2.00	1.00	-4.00	0.00	1.00	7.00
obj	-4.00	0.50	-1.00	0.00	0.00	0.00

and complete the next tableau:

	x_1	x_2	x_3	x_4	x_5	
obj						

What was the optimal solution found?

$x_1 =$ ☐ , $x_2 =$ ☐ , $x_3 =$ ☐ , $x_4 =$ ☐ , $x_5 =$ ☐ .

2. **(1 pt)** Apply the simplex method to show that $P(4.0)$ is unbounded (i.e., the objective function can be made as large as desired). Start at the following tableau

	x_1	x_2	x_3	x_4	x_5	
x_4	1.00	2.00	-3.00	1.00	0.00	5.00
x_5	2.00	1.00	-4.00	0.00	1.00	7.00
obj	-5.00	1.00	-1.00	0.00	0.00	0.00

and complete the next tableau:

	x_1	x_2	x_3	x_4	x_5	
obj						

Use the resulting tableau to exhibit an infinite family of feasible solutions of the form $x_i(t) = a_i t + b_i$ for $i = 1, \ldots, 5$, such that $\lim_{t\to+\infty} -5x_1(t) + x_2(t) - x_3(t) = +\infty$.

$a_1 = \boxed{}$, $b_1 = \boxed{}$,

$a_2 = \boxed{}$, $b_2 = \boxed{}$,

$a_3 = \boxed{}$, $b_3 = \boxed{}$,

$a_4 = \boxed{}$, $b_4 = \boxed{}$,

$a_5 = \boxed{}$, $b_5 = \boxed{}$.

3. **(0.5 pts)** Find the largest value of $\alpha = \boxed{}$ in the interval $[3.5, 4.0]$ for which the problem $P(\alpha)$ has a finite optimum. Justify your answer below.

Surnames	**Name**	**DNI**

Exercise 2: Integer Programming **(2.5 pts)**

1. **(1.5 pts)** Consider the following problem

   > A set of n jobs must be carried out on a single machine that can do only one job at a time. Each job j takes p_j hours to complete. Given job weights w_j for $j = 1, \ldots, n$, find the order in which the jobs should be carried out so as to minimize the weighted sum of their start times. All variables are positive integers.

   We define the decision variable x_j to denote the start time of job j, where $1 \leq j \leq n$. Express the fact that jobs should be done one at a time and complete the integer program below. Introduce new integer decision variables if needed.

   Minimize $\sum_{j=1}^{n} w_j x_j$
   subject to

   where $x_j \geq 0$ is an integer for $1 \leq j \leq n$.

2. **(1 pt)** A *vertex cover* of an undirected graph $G = (V, E)$ is a subset of the vertices $S \subseteq V$ such that each edge $e = \{u, v\} \in E$ satisfies that $u \in S$ or $v \in S$, that is, at least one endpoint of e is in S. Given a graph $G = (V, E)$ and a vertex weight function $w : V \to \mathbb{N}$, a *minimum vertex cover* of G is a vertex cover S of G having minimum total weight, that is, s.t. $\sum_{u \in S} w(u)$ is smallest among all vertex covers of G.

   In an attempt to find a 2-approximation algorithm for the Minimum Weight Vertex Cover problem, consider the following variation of the known 2-approximation for Vertex Cover.

input: graph $G = (V, E)$ and weight function $w : V \to \mathbb{N}$

$S \leftarrow \emptyset$

while S is not a vertex cover of G

  $C \leftarrow \{u \in V - S \mid \exists v \in V - S \;\; \{u, v\} \in E\}$

  select a vertex $u_{\min} \in C$ s.t. $w(u_{\min}) \leq w(u)$ for all $u \in C$

  $S \leftarrow S \cup \{u_{\min}\}$

return S

Does the above algorithm constitute a 2-approximation for the Minimum Weight Vertex Cover problem? Provide a proof or a counterexample to justify your answer.

Surnames	Name	DNI

Exercise 3: Heuristic Methods **(5 pts)**

A large company that produces huge amounts of data does not have the necessary resources to store monthly backups and decides to hire external backup providers. The company has O different offices and we know, for each office o, which is the amount of data b_o, in PetaBytes (106 GB), that it needs to store.

After studying several possibilities, a number of backup providers D are chosen. Using a backup provider d has a fixed cost f_d and an additional cost s_d for each PB it stores. Each backup provider d can store at most k_d PBs of data.

Each office can use just one backup provider, and each backup provider can store data from no more than 3 different offices. In addition, the total number of backup providers want to be kept limited to no more than 3.

The goal is to find out which backup centers will be used and how many PBs of each office will store every backup provider, so that the total cost is minimized.

The data is as follows:

$O = 1..7;$
$D = 1..5;$
$b = [200\ \ 200\ \ 200\ \ 200\ \ 200\ \ 200\ \ 200];$ // Demand per office (in PB)
$k = [1500\ \ 500\ \ 700\ \ 1000\ \ 1000];$ // Capacity per backup provider (in PB)
$f = [300\ \ 350\ \ 250\ \ 800\ \ 600];$ // Fixed cost
$s = [1.00\ \ 0.15\ \ 0.65\ \ 0.45\ \ 0.50];$ // Cost per PB

Because of the complexity of the optimization problem, we want to develop a heuristic algorithm. We are considering two options among the available in the literature, Greedy and GRASP.

1. **(2.5 pts)** Specify the algorithm for the Greedy and GRASP constructive phase, including the candidates, the greedy function $q(\cdot)$, and the equation describing the RCL. Specify $q(\cdot)$ using mathematical notation and a short descriptive text.

2. **(2.5 pts)** Let us assume that the algorithm specified in Exercise 3.1 for the Greedy is being executed. Complete the following table: for each iteration, compute the value of the proposed greedy function $q(\cdot)$ for all the candidates.

	Office						
Iter # 1	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 2	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 3	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 4	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 5	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 6	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

	Office						
Iter # 7	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$q(o)$							
d							

Detail the obtained solution in the next table.

Backup Provider	**Office** 1	2	3	4	5	6	7
1							
2							
3							
4							
5							

What is the total cost?

Do you think this solution is optimal? Why?