Operations Research

Mock Midterm Exam

October 29, 2020

1. With the start of school approaching, a store is planning on having a sale on school materials. They have 600 notebooks, 500 folders and 400 pens in stock, and they plan on packing it in two different forms. In the first package, there will be 2 notebooks, 1 folder and 2 pens, and in the second one, 3 notebooks, 1 folder and 1 pen. The price of each package will be €6.50 and €7.00, respectively. How many packages should they put together of each type to obtain the maximum revenue? (10)

2. Consider linear programming problem

$$\text{maximize } z = x_1 + 2\,x_2$$

subject to

$$\begin{gathered} -2\,x_1 + x_2 + x_3 \leq 2\,, \\ -x_1 + x_2 - x_3 \leq 1\,, \\ x_1, x_2, x_3 \geq 0\,. \end{gathered}$$

   (a) Write this LP in the standard form.

   (b) Perform the simplex algorithm until you reach a stopping point. Note that the final tableau corresponds to an unbounded solution.

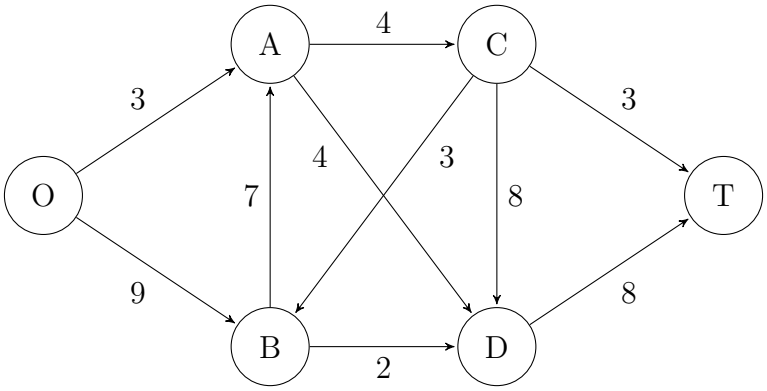
   (c) Find a basic feasible solution for which the objective function takes a value of at least $z = 1000$.

   (d) Write out the *dual* problem to this LP.

   (e) Show that the feasible region of the dual is empty.

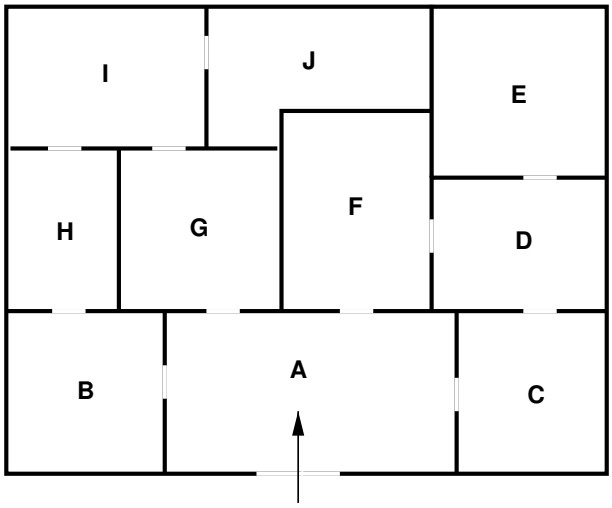
(3+7+3+4+4)

3. Find the minimum cut in the network below. Describe in words how you found the solution.



(10)

4. A museum director must decide how many guards should be employed to control a new wing. Budget cuts have forced him to station guards at each door, guarding two rooms at once. Formulate a linear program to minimize the number of guards, referring to the map below. (No need to solve the problem on the exam.)



(10)

5. Consider the Pyomo program attached.

   (a) Write out the problem that is being solved in mathematical notation.

   (b) Describe what the program does in plain language. You may choose a concrete application scenario beyond what is implied by the names in the code.

   (c) What happens if you change the declaration of the decision variables from `within=Boolean` to `within=NonNegativeReals`?

(5+10+5)

In [1]:
```
from pyomo.environ import *
from pyomo.opt import *
opt = solvers.SolverFactory("glpk")
```

In [2]:
```
Jobs  = [0, 1, 2, 3, 4]
Slots = [0, 1, 2, 3, 4]

Due     = [2, 2, 3, 1, 3]
Penalty = [19, 12, 34, 30, 22]

model = ConcreteModel()

model.a = Var(Jobs, Slots, within=Boolean)

model.z = Objective(
    expr = sum(model.a[j,s]*(s-Due[j])*Penalty[j]
               for j in Jobs for s in Slots if s>Due[j]),
    sense=minimize)

def all_jobs_assigned_rule (model, j):
    return sum(model.a[j,s] for s in Slots) == 1

model.jobs = Constraint(Jobs, rule=all_jobs_assigned_rule)

def all_slots_assigned_rule (model, s):
    return sum(model.a[j,s] for j in Jobs) == 1

model.slots = Constraint(Slots, rule=all_slots_assigned_rule)

results = opt.solve(model)
```

In [3]:
```
model.a.get_values()
```

Out[3]:
```
{(0, 0): 1.0,
 (0, 1): 0.0,
 (0, 2): 0.0,
 (0, 3): 0.0,
 (0, 4): 0.0,
 (1, 0): 0.0,
 (1, 1): 0.0,
 (1, 2): 1.0,
 (1, 3): 0.0,
 (1, 4): 0.0,
 (2, 0): 0.0,
 (2, 1): 0.0,
 (2, 2): 0.0,
 (2, 3): 1.0,
 (2, 4): 0.0,
 (3, 0): 0.0,
 (3, 1): 1.0,
 (3, 2): 0.0,
 (3, 3): 0.0,
 (3, 4): 0.0,
 (4, 0): 0.0,
 (4, 1): 0.0,
 (4, 2): 0.0,
 (4, 3): 0.0,
 (4, 4): 1.0}
```

In [4]:
```
model.z.expr()
```

Out[4]:
```
22.0
```