W1005 - Fall 2014
Homework 6

- Due by Friday 4pm (Dec 5th).
- See submission instructions.
- Always include your name and UNI at the top of your submitted files.

1. The following questions should be answered in the text file **hw6.txt** (word/pdf file is OK too).
   Generate a vector of random numbers by executing two commands:
   ```
   >> rng(10)
   >> V = floor(rand(1,8) *51)
   ```

   a) Show the steps that the *Mergesort* algorithm takes to sort V in *ascending* order (see slide #33 for an example). You only need to show V after each 'merge' step.

   The following question refers to the *Quicksort* algorithm (with the partitioning strategy & approach for equal elements described in class), and slides #44-45 in particular.

   Generate a vector of random numbers by executing two commands:
   ```
   >> rng(3)
   >> V = floor(rand(1,9) *50)
   ```

   b) Show the steps that the algorithm takes to sort V in *ascending* order where the *first* element is always chosen as the pivot.
   Note: you need to show how you recursively sort each group

2. Implement "indirect-sorting" with the insertion sort algorithm. Name your function **obj_sort.m**. Your function is going to sort "objects" of some unknown type.

   a) You should start with the isort.m function (from hw5, see solution), and modify it as necessary (any changes to code or template are allowed).

b) Your function **obj_sort** takes as input a cell array 'R' (each element of the array is an "object"), and a function handle 'cmp'.
c) Instead of returning the sorted array R, you will return a vector V with the indices of the elements in the proper order (according to their sorted order).
d) To compare two "objects", use the '**obj_cmp.m**' function from hw5.
e) You should assume that there are no errors in R. That is, all the objects stored in the array have the same (unknown) type.

3. Recall the "write a best seller" problem from class (optimization notes). We will solve almost an identical problem which differs from WBS as follows:

   A. Instead of a single quality report, you now have two of them {R1,R2}. R1 consists of K1 marks, and R2 of K2 marks for each word in the vocabulary.

   B. You are also given two vectors {B1,B2} describing the total quality for each report respectively, instead of one (B).

   Obviously, the 'assumptions' and 'approach' described in class are almost the same for this "WBS" problem.

a) Formulate the problem WBS2 in mathematical notation. You can include a separate text file (**wbs2.txt**) with your code, or just use block comments in the function below.
b) Convert your notation from part (a) to solver form. Hint: you cannot just copy from the lecture slide, there is a difference.
c) Write a function named '**wbs2.m**'. Your function should have one input parameter 'specs', a struct with the fields {N,C,R1,R2,B1,B2}.
d) Assume that there are no errors in the fields {N,C}. Add a check to ensure that the matrices/vectors supplied in fields {R1,R2,B1,B2} have the right size. The number of marks {K1,K2} can be any positive integer, but {R1,B1} should be consistent with each other (same goes for {R2,B2}).
e) Your function should implement the conversion of the input parameters to solver form, and then make two calls to the solver:
   A. x1 = *linprog*(f,A,b,Aeq,beq,lb,ub)

B. x2 = *linprog*(problem)

f) For call (B), make sure to read the browser documentation. You should be passing a struct 'problem' which has appropriate fields.
g) Check that the solutions you obtained {x1,x2} are identical.

   Note: the problem is much easier if you first review the lecture slides carefully.

   Note: it is not difficult to verify that your code is working by generating random data {C, R1,R2} specifying some small values of N and adjusting {B1,B2}.

4. Team Results in European Cup Matches (part 2):

   a) Recall problem 4 from HW5. Here we modify our code slightly as follows:

      Write a program '**get_match2.m'** that in addition to the previous code, replaces all the hyphens ('-') in your strings with underscores ('_'). This can be useful since strings with hyphens ('-') cannot be field names in a struct.
      Hint: Use ***regexprep*** function
      Hint: instead of @*lower* use call to regexprep as an *anonymous* function.

Your zip folder should include the following files:
wbs2.m (wbs2.txt is optional)
hw6.txt
obj_sort.m
get_match2.m