Exam 1 Format (Tue. 2/27)

The exam is 75 Minutes, open book, open notes, but you must bring the notes in paper format. No computer, tablet, or smart phone is allowed. For calculations, you can use a handheld calculator, or leave the answer in a format that is easily computable by Python.

1. Multiple Choice

3 Multiple choice questions asking you to define or apply any of the following concepts.

- **Basic optimization**: decision variables, objective, constraints, domain, local minimum, local maximum, global minimum, global maximum, enumeration, how objective changes when we add or remove constraints (see homework 1).
- **Basic probability**: outcomes, event, probability of an outcome, probability of an event, mutually exclusive events, random variable (RV), discrete RV, continuous RV, mean, expected value, variance, standard deviation, covariance.
- **Relationship between RVs**: independence, correlation, joint probability distribution, conditional probability, Bayes' theorem.
- **Probabiltiy distributions**: cumulative distribution function (CDF), probability mass function (PMF), probability density function,
- **Decision trees**: outcome nodes, decision nodes, event nodes, how to solve a given tree, assumptions involved in a decision tree analysis.
- **Programming basics**: CPU, memory, input/output, what can computers easily do, what can't they easily do.
- **Python concepts**: objects, variables, if statement, for loop, lists, dictionaries, numpy array, pandas.

2. Analyzing a case using decision tree

A text problem in which you will be asked to recommend a decision by constructing and solving a decision tree, similar to those in homework 2 and 3.

Points to pay attention to:

- The order of nodes in the tree (must following the order of information revelation. For example, whenever there is a decision node, all of the previous outcomes above the node must be known with certainty.)
- May need to apply conditional probability and Bayes' rule (or equivalently using the joint probability table as covered in class) to solve for probabilities.
- Computing the value of each outcome carefully.
- Solving an event node by taking the expected value of the children.
- Solving a decision node by taking the best value of the children.

3. Solving a problem involving probability.

The problem will have 2-3 parts, asking you to perform calculations similar to those in homework 2 and 4.

Concepts in which you may be asked to apply:

- common formula governing probabilities (DMD 2.2).
- independence, correlation.
- conditional probability and Bayes' rule.

- the total law of probability (DMD 2.12).
- CDF, PMF, PDF and calculations involving them.
- computing the mean, variance and standard deviation of a discrete random variable.
- formula for expectation of linear combination of random variables (linear combination of X and Y means sums of the form $aX + bY$ where a and b are two constants).
- formula for the variance and standard deviation of linear combination two correlated random variable.
- Central Limit Theorem and the Normal approximation.
- Bernoulli, Binomial, Normal, and Uniform distributions.
- Stating and critically analyzing the assumption behind each analysis you do.

4. Python coding

There are two parts: one asks you to predict the output of a given segment of Python code, another asks you to insert a few lines to complete a segment of Python code.

Python constructs covered:

- **Assignment**: variable assignment, multiple assignment, concept of scope of variables.
- **Conditional statements**: `if`, `elif`, `else`.
- `for` **loops**, both iterating through a `range` and iterable object like a `list` or `numpy` array. `continue` and `break`.
- **Strings**: `print`, `str.format`,`str.split`,`str.join`.
- **Lists**: constructing a list explicitly or through list comprehension, slicing a list, `list.append`, `list.index`.
- **Range-like lists**: `range`, `np.arange` (with the arguments start, end, and skip), `np.linspace`.
- **Functions**: defining a function. Adding docstring.
- **scipy.stats**: declaring distributions using `norm`, `uniform`, `binom`, `geometric`, `poisson`, `expon`. Calculating statistics using `mean`, `std`, `var`, `cdf`, `pdf`, `pmf`. Sampling using `rvs` (both a single number and an array of numbers).
- **matplotlib.pyplot**: `plot`, `hist`, `show`, `title`, `xlabel`, `ylabel`, `vlines`, `xlim`, `ylim`, `legend`.
- **numpy**: declaring an array (both from a list as well as a new array of zeros or ones), array indexing (explicitly, fancy indexing, by boolean mask), array slicing, assigning value to a slice, element wise arithmetics, `shape`, `sum`, `average`, `sqrt`, `power`,`log`, `abs`, `sort`, `argsort`, `max`, `min`, `maximum`, `minimum`, `hstack`,`vstack`,`T`,`argmax`,`argmin`,`argwhere`, `cumsum`, `copy`,`loadtxt`,`savetxt`.
- **pandas**: explicitly defining a Series or DataFrame, `read_csv`, `read_excel`, `to_csv`, `as_matrix`, `info`, slicing, filtering.