Project Description (*Due on last day of class*)

This description provides detailed information about the group project that you are to submit as part of your course work. Each group consists of at most two students, and is responsible of developing a project proposal, presentation slides, and final technical project report.

1 Project Proposal

The project proposal is mainly intended to make sure you decide on a project topic and get feedback from course instructor early. As long as your proposal follows the instructions below and the project seems to have been thought out with a reasonable plan, you should do well on the proposal.

- Due by one week after the midterm exam.
- The proposal should be no more than 2 pages excluding the references.
- The project proposal should provide a basic description of your project, why it is interesting, and why you have chosen to implement it in a particular way.
- Below is an outline for the proposal to be used as a guideline, but as long as you answer these basic issues, you can organize the report whichever way makes sense to you.
  - Title, Author(s)
  - Project abstract
  - Project objectives
  - Proposed methodology
  - Implementation plan and schedule
  - Related works

2 Final Report

The report should be written professionally with embedded figures and tables (such as illustrative flow charts and data plots). Reports done by groups should contain both names of both members; each member of the same group will receive equal grades.

- **Due: last day of class.** No Late submission accepted.
- Your final report should use the arxiv-style template.
  https://github.com/kourgeorge/arxiv-style
- You should include a brief statement on the contributions of different members of the team in the report. Team members will normally get the same grade, but we reserve the right to differentiate in egregious cases.
- The following is a suggested structure for the report:
  - Title, Author(s).

- **Abstract**
- **Introduction**: This section introduces your problem, and the overall plan for approaching your problem.
- **Background/Related Work**: This section discusses relevant literature for your project.
- **Approach**: Algorithms used or developed.
- **Theoretical results (if relevant)**: Include assumptions, proof sketches.
- **Experiment results (if relevant)**: Details on experiments done. The goal is to describe in enough detail that the results are reproducible.
- **Conclusion**: What have you done and learned? Suggest future ideas.
- **References**

3 Recommended References

Part I: Applications of RL to solve OR/IE problems.

1. Queueing systems
   - "Reinforcement Learning for Optimal Control of Queueing Systems",
   - "Stable Reinforcement Learning with Unbounded State Space": unboundedness of queueing networks.
   - "Queueing Network Controls via Deep Reinforcement Learning": advanced RL algorithm & variance reduction methods.
   - "An Online Learning Approach to Dynamic Pricing and Capacity Sizing in Service Systems": simpler model & algorithm, more theoretical insights.
   - "Integrating learning and adaptive control in queueing systems with uncertain payoffs"
   - "Learning unknown service rates in queues: A multiarmed bandit approach"
   - "On Learning the $c\mu$ Rule in Single and Parallel Server Networks"
   - "Online Learning and Pricing for Service Systems with Reusable Resources"
   - "Staffing of Time-Varying Queues to Achieve Time-Stable Performance"
   - "The Value of Dynamic Pricing in Large Queueing Systems"
2. Inventory management
   - On implications of demand censoring in the newsvendor problem
   - "Marrying Stochastic Gradient Descent with Bandits: Learning Algorithms for Inventory Systems with Fixed Costs"
   - "Online Advance Scheduling with Overtime: A Primal-Dual Approach"
   - "Optimal Learning Algorithms for Stochastic Inventory Systems with Random Capacities"
   - "Closing the Gap: A Learning Algorithm for the Lost-Sales Inventory System with Lead Times"
   - "Advance Scheduling with Personalized Learning"
   - "Tailored Base-Surge Policies in Dual-Sourcing Inventory Systems with Demand Learning"

3. Revenue management
    - “Dynamic Bid Prices in Revenue Management”
    - “Nonparametric Pricing Analytics with Customer Covariates”
    - “On the (Surprising) Sufficiency of Linear Models for Dynamic Pricing with Demand Learning”
    - Dynamic Pricing Without Knowing the Demand Function: Risk Bounds and Near-Optimal Algorithms
    - “Dynamic Pricing with Financial Milestones: Feedback-Form Policies”
    - “Sample-based optimal pricing”
4. Healthcare
    - “Personalized Hospital Admission Control: A Contextual Learning Approach”
    - “Inpatient Overflow: An Approximate Dynamic Programming Approach”
    - “Personalized HeartSteps: A Reinforcement Learning Algorithm for Optimizing Physical Activity”
    - “Rapidly Personalizing Mobile Health Treatment Policies with Limited Data”
    - “Data-Driven Appointment-Scheduling Under Uncertainty: The Case of an Infusion Unit in a Cancer Center”
5. Others
    - Making the most of your day: online learning for optimal allocation of time
    - OR-Gym: A Reinforcement Learning Library for Operations Research Problems
    - Reinforcement Learning in Healthcare: A Survey

Suggested roadmap of Part I

- Find a problem you are interested in.
- Formulate an MDP model; develop the RL algorithm.
- Test your algorithm with simulation or real data.
- Compare with existing methods

Part II: RL methodologies.

1. Online RL: The agent are able to gather new data by current policy but in many applications the data acquiring process is time consuming and expensive.
    - Off-policy policy optimization and evaluation
        - “Reliable O-policy Evaluation for Reinforcement Learning”
        - “Divergence-Augmented Policy Optimization: Constrains on state-action visitation frequency”
        - “Multiple behavior policy: Importance Sampling Policy Evaluation with an Estimated Behavior Policy”
        - DICE-type algorithms: “Off-Policy Evaluation via the Regularized Lagrangian”

- Model-based RL
  - MuZero Model based AlphaZero: "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model"
- Distributional RL
  - "Distributional Reinforcement Learning for Efficient Exploration"
  - Risk-sensitive control: "Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy & Stochastically Dominant Distributional Reinforcement Learning"

2. Offline RL: How to perform reinforcement learning with a fixed dataset and (very possibly) insufficient observation?

   - "Offline Reinforcement Learning - Tutorial, Review, and Perspective on Open Problems": overview
   - "MOPO: Model-based Offline Policy Optimization": model-based method & pessimism
   - "Exponentially Weighted Imitation Learning for Batched Historical Data": advantage reweighed imitation learning, policy improvement from fixed-dataset
   - "Divergence Augmented Policy Optimization": policy-based method, can be extended to Offline setting.
   - "Provably Good Batch Reinforcement Learning Without Great Exploration"
   - "The Importance of Pessimism in Fixed-Dataset Policy Optimization": theoretical justification of pessimism in offline RL

Suggested roadmap of Part II

- Choose a topic provided you find interesting.
- Select a reference paper or find the reference material on your own as your project baseline.
- Develop an improved algorithm and/or theoretical justification.
- Test your algorithm with simulation data/real world data.