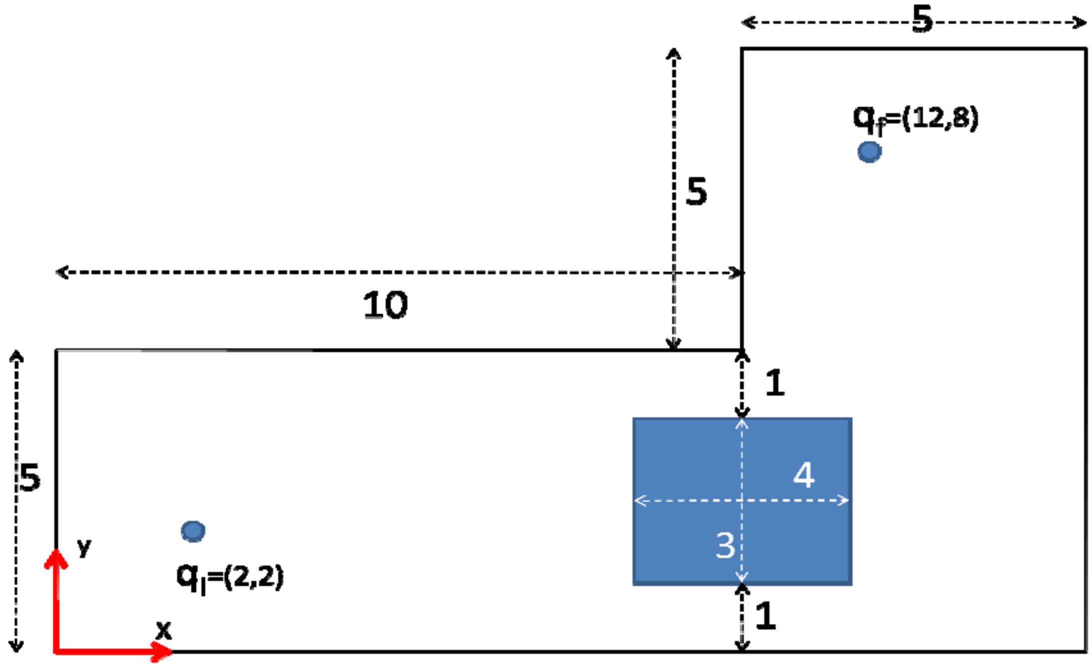
ME 131: Homework #1
(due Wed. April 15, 2009)

This homework focuses on sample based planners (Chapter 5 of the Lavalle Book). Consider the motion planning environment below, consisting of an L-shaped room with a rectangular obstacle symmetrically located in the "neck" of the L. For convenience a reference coordinate system is placed in the lower left hand corner of the room. The goal is to find a collision-free path for a point-like robot from the initial configuration (2,2), as measured in the reference system, to the goal location (12,8).



You should implement (in any programming environment that you choose) the bare bones of sample-based planner to solve this problem. As seen in Chapter 5 of the Lavalle book, there are *many* different types of sampled based planners. However, to keep things focused, you should consider implementing the *Rapidly Exploring Dense Tree* (RDT) algorithm(s) of Section 5.4, or a *K-nearest neighbor roadmap* discussed in Section 5.6 (sampled based planners for multiple queries). No matter which approach you choose, you should implement the following subroutines:

1. ***Collision check.*** Construct a procedure to check if a candidate point (described by 2D Cartesian coordinates) is infeasible—i.e., located inside the obstacle or outside of the room. Describe the principle behind your collision checking procedure.
2. ***Sample generation procedure.*** While this procedure will vary with the type of sample based planner you choose, you must develop a procedure to generate new samples configurations. For the k-nearest neighbor approach, you need to generate k random points near an existing vertex in the search graph. For the RDT approach, you must

generate a sequence of samples that will be connected to the evolving search tree. You may use one of the sample generation strategies discussed in Section 5.2, or devise your own.

3. ***Local Planning Method (LPM).*** This procedure connects newly generated sample points to existing points/vertices in the search graph via a local path. In an actual implementation on a given robot, this procedure must take the robot's motion model into account. For this homework, it is perfectly fine to use a straight line path to connect vertices. Note that in the k-nearest-neighbor planner, you want to try to connect a new sample point to several other sample points and existing search graph vertex points, while in the RDT method, you need to determine the closest point on the existing RDT graph. Note that the LPM must check if the path intersects an obstacle. This can be practically implemented by sampling the straight line path at uniform intervals, and use the collision check procedure at these points on the path. A more sophisticated path check approach is welcome, but not necessary.
4. ***A search termination criterion.*** Sample based planners do not necessarily construct a path that exactly reaches the goal. Typically, they conclude using one of these approaches:
    a. Epsilon-ball criterion. Does there exist a vertex in the constructed search graph (or more generally, a point on one of the constructed local paths) within a distance epsilon of the goal? If so, then the goal is assume to be reached.
    b. Direct goal connectivity. Can the LPM construct a collision free path from an existing graph vertex (or point) to the goal? If so, construct the path to the goal and then conclude with success.

To demonstrate your planner, show the sampled points and their connecting paths generated up until the time that the goal is reached. You need not implement the detailed graph search algorithm that will determine a specific path sequence to the goal. An 10% extra credit will be applied to those homeworks that also implement and demonstrate a graph search component as well.