Tutorial for Week 6

See Russell and Norvig, chapter 5

1. In the context of CSP search, explain why it is a good heuristic to choose the variable that is **most** constrained, but the value that is **least** constraining.

2. Give a formulation of the following as a CSP:
   class scheduling: there is a fixed number of teachers and classrooms, a list of classes to be offered and possible time slots for each class. Each teacher has a set of classes he or she can teach. (There are many different formulations possible here.)

3. Consider this cryptarithmetic problem:

```
   C O U N T
 -   C O I N
   ---------
     S N U B
```

   Give a CSP formulation of the problem. The letters correspond to single distinct digits, and there are no leading zeros. You should use variables to express any auxiliary values; assume the constraint language allows arithmetic constraints like $a.X + b.Y = c.D + d$ for integer constants a, b, c, d.

4. In the Australia map colouring CSP, show that the AC-3 algorithm can detect that the partial assignment $\{WA = red, V = blue\}$ cannot be extended to find a solution to the problem.

   The problem assigns one of three colours to each state so that no two adjacent states have the same colour. The adjacent states are as shown in figure 1.

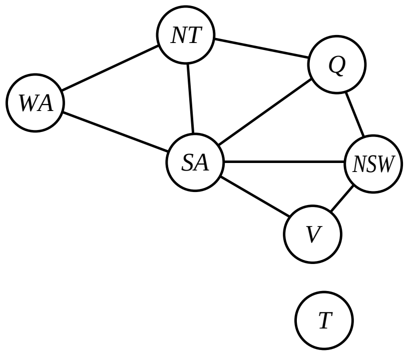

Figure 1: Constraint graph for colouring problem

```
function REMOVE-INCONSISTENT-VALUES( Xi, Xj) returns true iff we remove a value
   removed ← false
   for each x in DOMAIN[Xi] do
      if no value y in DOMAIN[Xj]
               allows (x,y) to satisfy the constraint between Xi and Xj
         then delete x from DOMAIN[Xi];  removed ← true
   return removed
```

```
function AC-3( csp) returns the CSP, possibly with reduced domains
   inputs: csp, a binary CSP with variables {X1, X2, ..., Xn}
   local variables: queue, a queue of arcs, initially all the arcs in csp

   while queue is not empty do
      (Xi, Xj) ← REMOVE-FIRST(queue)
      if REMOVE-INCONSISTENT-VALUES(Xi, Xj) then
         for each Xk in NEIGHBORS[Xi] do
            add (Xk, Xi) to queue
```