

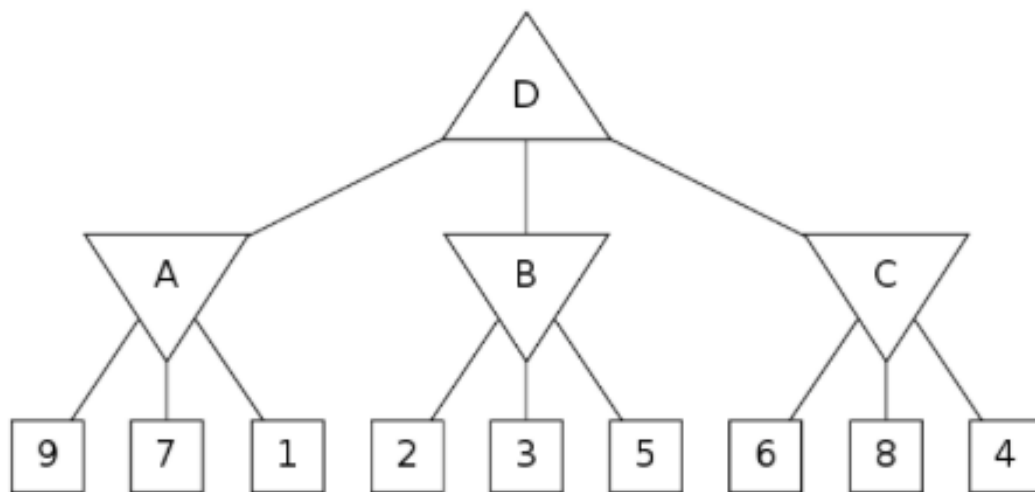
人工智能导论作业 2

总分 88 分。

课堂提交时间 11 月 3 日，提交纸板（手写和打印都可以）。

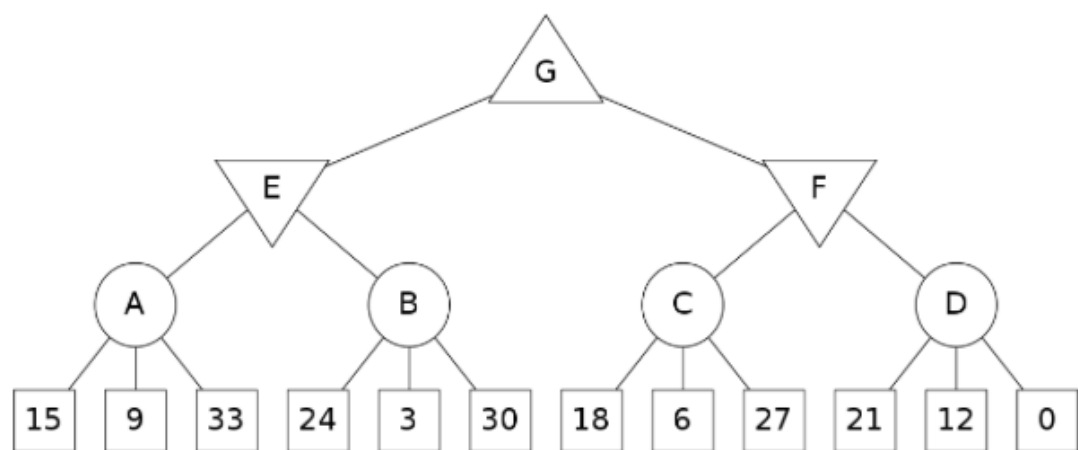
问题 1， 最小最大 MINIMAX（4 分）

以下是一个零和游戏博弈树。正三角代表最大化利益玩家的选择，倒三角代表的是最小化利益玩家选择。最大化利益玩家的游戏结果值列在树叶节点方框内。假设两个玩家都是选择最优化的行动，那么利用最小最大搜索算法，给出 A，B，C，D 的值。



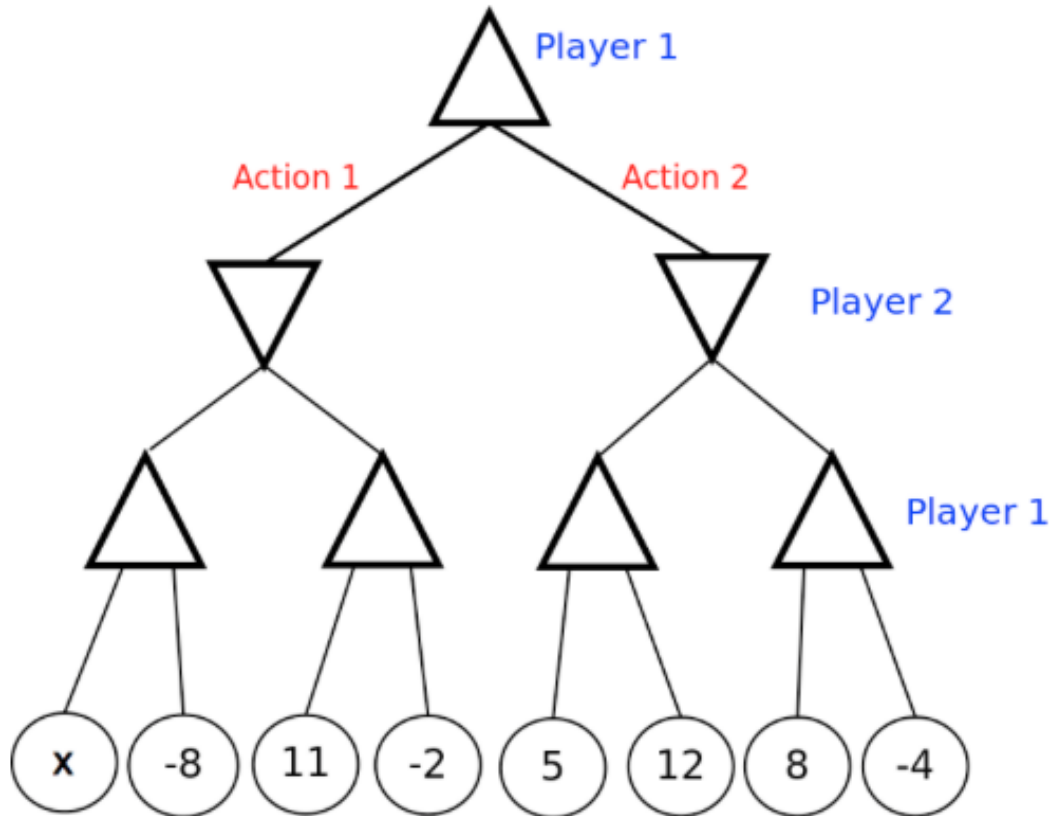
问题 2， 期望最小最大 EXPECTIMINIMAX（7 分）

以下的游戏博弈树包含最大值节点（正三角），最小值节点（倒三角），和机遇节点（圆圈）。机遇节点下每个可能的行动是相同概率发生的。树底部叶节点（方框）中给出了相对于根节点（最大利益化玩家）的游戏结果值。假设两个玩家都是采取最优化的行动，那么利用期望最小最大搜索算法，请给出节点 A, B, E, C, D, F, G 的值。



问题 3，未知的叶结点值（4 分）

在以下的博弈树中，有一个叶节点的值未知（标记为 x ）。玩家 1 先走，并试图最大化博弈结果值。



对以下 3 个问题给出 x 值的约束条件（即 x 值可能的取值范围，使用整数和 $>$, $<$ 符号）。如果 x 的值没有符合条件的，则写明 x 的范围为没有。

3-1) 假设玩家 2 试图最小化博弈结果值（玩家 1 知道这个情况）。 x 取什么值时，玩家 1 能保证选择 Action 1 的方向？

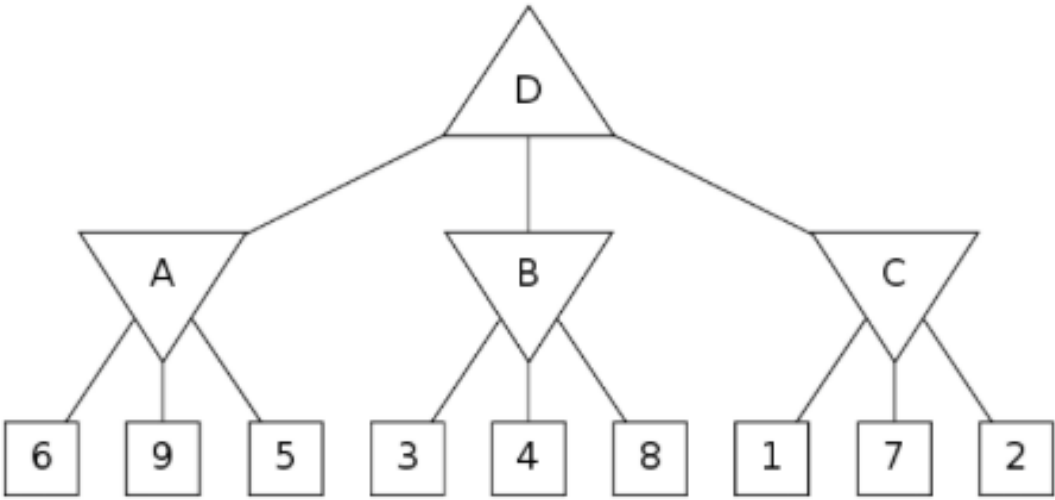
3-2) 假设玩家 2 随机（相等概率）地选择一个行动方向（玩家 1 也知道玩家 2 的这个策略）。那么， x 取什么值时，玩家 1 能保证选择 Action 1 的方向？

3-3) 对于以上博弈树，当玩家 1 是最大化者，玩家 2 是最小化者，这棵树的最小最大值定义为根节点的值；当玩家 1 是最大化者，而玩家 2 随机选择（相同概率下）行动方向，这棵树的期望最大值定义为根节点的值。问 x 为何值时，这个树的最小最大值能够大于这个树的期望最大值？

3-4) 问：是否有这样一个博弈游戏，使得其最小最大值严格大于其对应的期望最大值？

问题 4， Alpha-Beta 剪枝（13 分）

在以下的游戏博弈树中，包括游戏值最大化玩家（正三角），最小化玩家（倒三角）。假设两个玩家都采取最优化的行动方案，请使用 **alpha-beta** 剪枝来找到根节点的值。搜索是从左到右进行，子节点访问的顺序是选择最左边未被访问过的子节点。



在下面给出节点 A,B,C,D 的值；如果相应的叶结点被剪掉了，请在以下列表中叶结点值所对应的节点旁标记一个叉“x”。

节点值：

A =

B =

C =

D =

剪枝节点：

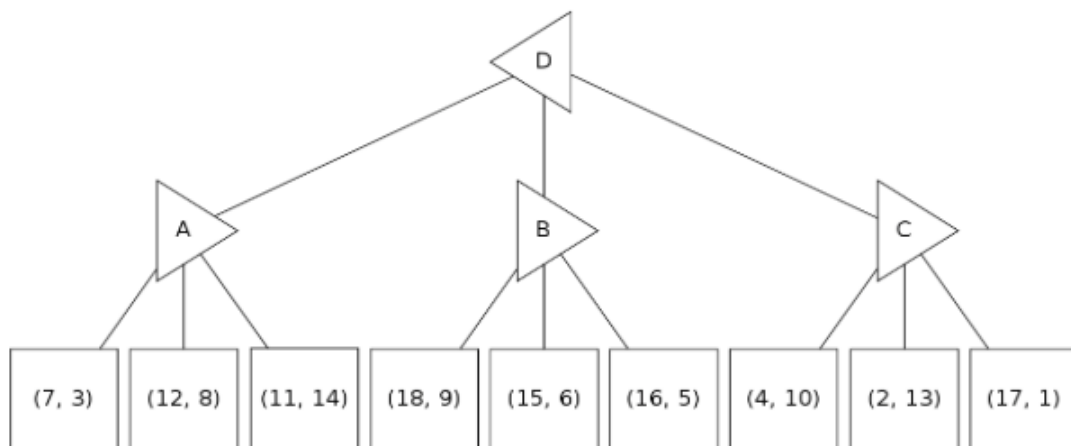
节点 6	节点 9	节点 5
节点 3	节点 4	节点 8
节点 1	节点 7	节点 2

问题 5，非零和游戏（11 分）

在零和游戏中，在所有的终局状态 s 里，玩家 A（最大化者）和玩家 B（最小化者）的利益值之和为 0，即 $U_A(s) + U_B(s) = 0$ 。在这种情况下， $U_A(s) = -U_B(s)$ ，所以我们可以认为玩家 B 是在最小化玩家 A 的利益值。标准的最小最大值搜索算法即是用来计算零和游戏的最差情况下的状态节点值。在以下问题里我们将考虑非零和游戏情况，即两个玩家的利益之和不一定是 0。叶结点状态值现在表示为 (U_A, U_B) 。玩家 A 将最大化 U_A （括号里的第一个数），玩家 B 将最大化 U_B （括号里的第二个数）。

在以下的非零和游戏博弈树里，左指向的三角对应玩家 A，它将最大化括号里的第一个利益值，而右指向的三角代表玩家 B，它将最大化括号里的第二个利益值。利益值将从树的底层向上传递到树根节点，应用的是普通化的最小最大值算法。如果遇到利益值相同的情况，总是选择最左边的子节点。

5-1) (8 分) 请为节点 A, B, C, D 选择正确的利益值对。你的答案应该是 (X, Y) 的形式，X 是玩家 A 的值，Y 是玩家 B 的值。



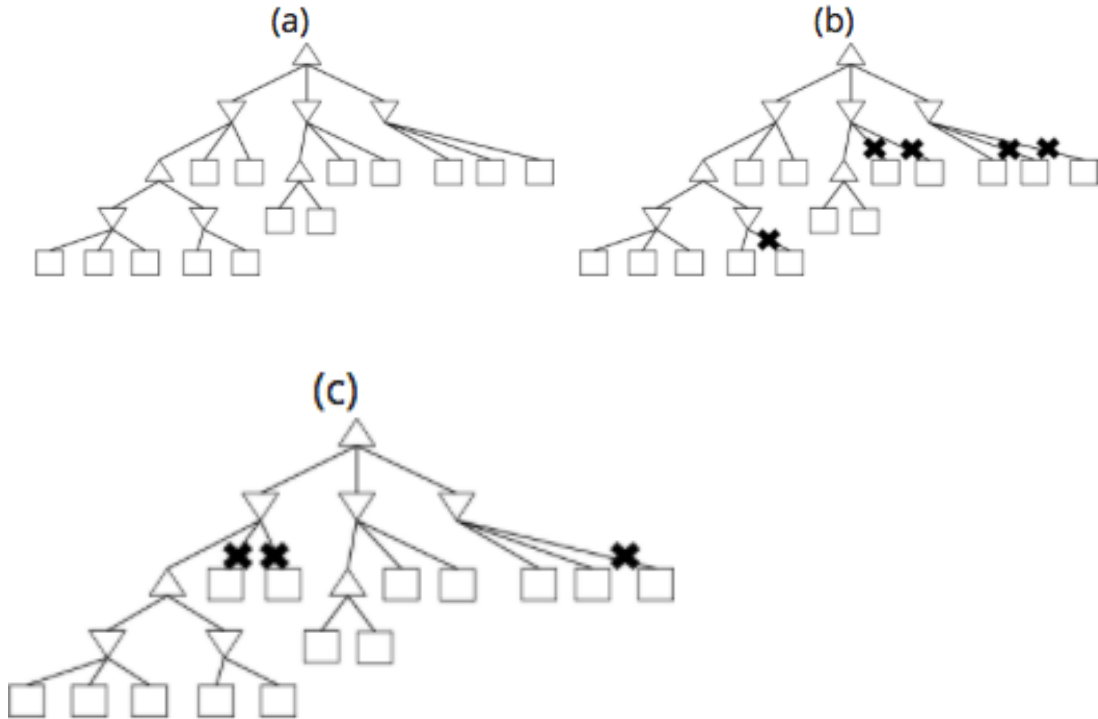
A =
B =
C =
D =

5-2) (3 分) 假设通过普通化的最小最大值算法计算的根节点的值是 (U_A, U_B) ，并且在所有终局节点里 A 和 B 的利益值都不同。问以下的哪些论断是对的？

- a) 假设玩家 A 和 B 都选择最优化的方向来行动，那么在终局时，玩家 A 的利益值可以保证就是 U_A 。
- b) 假设玩家 A 和 B 都选择最优化的方向来行动，那么在终局时，玩家 B 的利益值可以保证就是 U_B 。
- c) 假设玩家 B 的行动决策不是最优化的，但是玩家 A 的是最优化的，那么玩家 A 的利益值在终局时能保证至少是 U_A 。

问题 6，可能的剪枝（3 分）

假设我们在下图（a）中的博弈树上运行 alpha-beta 剪枝，从左向右扩展子节点。



问以下论断哪些是正确的？

- 1) 在图（a）中存在一组对终局节点（方块节点）的赋值，使得没有一个树枝被剪掉。
- 2) 存在一组对终局节点（方块节点）的赋值，使得剪枝情况如图（b）所示可以发生。
- 3) 存在一组对终局节点（方块节点）的赋值，使得剪枝情况如图（c）所示可以发生。
- 4) 以上论断都不对。

问题 7，次优化策略（7 分）

假设玩家 MAX 和玩家 MIN 在玩一个有限回合的零和博弈游戏。MAX 通过最小最大算法计算的树根节点的值是 M 。

假设每个玩家在每一轮都至少有 2 个可能的行动方向。而且，不同行动路线导致不同的终局节点值。

7-1. 问，以下的哪些论断是对的？

a) 假定玩家 MIN 在每一轮的行动都是次优化的（不是最优的），但是玩家 MAX 不知到玩家 MIN 的策略是次优化的。那么这个博弈游戏的终局结果值可能会大于 M （对 MAX 更有利）。

b) 假定玩家 MIN 在每一轮的行动都是次优化的（不是最优的），如果玩家 MAX 按照最小最大值策略来选择其行动，那么游戏的终局结果值可能会小于 M 。

7-2. 这里假设玩家 MIN 的行动选择在每一轮都是随机性的（其概率是均匀分布的），并且玩家 MAX 知道这一点。问，以下的那些论断是对的？

a) 对于玩家 MAX 来说，存在一种策略能够使得玩家 MAX 的终局结果值一定大于 M 。

b) 对于玩家 MAX 来说，存在一种策略能够使得玩家 MAX 的期望终局结果值好于 M 。

c) 如果玩家 MAX 要最大化它自己的期望终局结果，那么玩家 MAX 应该使用最小最大值策略（换句话说，这种最小最大值策略是假设玩家 MIN 的行动策略是最优化的）。

7-3. 问，以下的那些论断是对的？

a) 假设玩家 MIN 在每一轮的行动是次优化的，那么玩家 MAX 遵循最小最大值策略行动能够保证其终局结果值要大于 M 。

b) 假设玩家 MIN 在每一轮的行动是次优化的，并且玩家 MAX 知道它的对手行动是次优化的，那么存在一种策略使得玩家 MAX 能够在终局取得一个大于 M 的结果值。

问题 8， 课程安排（7 分）

计算机课程被安排在每周一，三，五。总共有 5 门课程，3 名教师，每名教师在一个时间只能教一门课。如何安排课程的教师？

- 课程包括：
 - 课程 1 – 计算机编程简介， 时间 8: 00 – 9: 00AM
 - 课程 2 – 人工智能导论，时间 8: 30 - 9: 30AM
 - 课程 3 – 软件工程， 时间 9: 00 – 10: 00 AM
 - 课程 4 – 计算机视觉，时间 9: 00 – 10: 00 AM
 - 课程 5 – 机器学习， 时间 10: 30 – 11: 30AM
- 教师包括：
 - 教师 A， 能够教课程 1， 2， 5
 - 教师 B， 能够教课程 3， 4， 5
 - 教师 C， 能够教课程 1， 3， 4

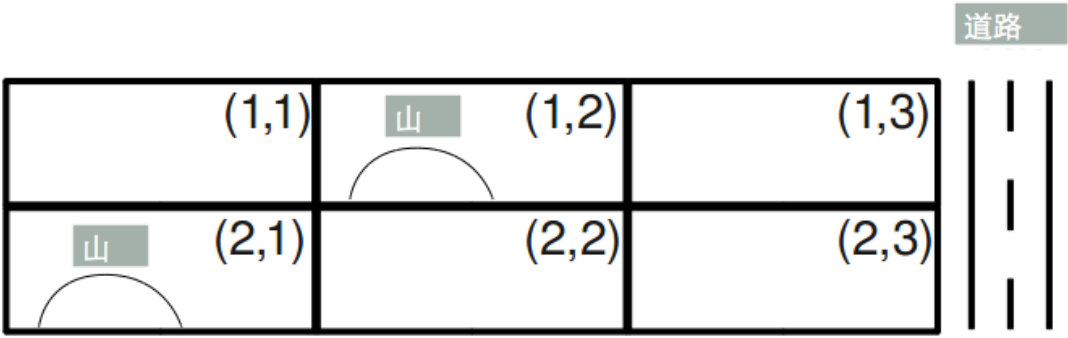
8-1) (4 分) 把以上问题描述为一个约束满足问题（用变量，值域和约束来描述。提示：一门课程是一个变量），指明变量的值域和约束；约束可以是隐式表达形式。

8-2) (2 分) 画出相应的约束图？

8-3) (1 分) 这个约束图应看上去近似树状。找到一个相应的切集（去掉它们，剩余的图变为树结构），并解释为什么将利用树状约束图能对解决这个问题有所帮助？

问题 9，校园规划问题（32 分）

假设由你来决定一个校园中新区域的规划设计。这个新区域中有四个建筑：一个行政管理楼（A），一个汽车站（B），一个教室楼（C), 和一个学生宿舍楼（D）。这些建筑必须坐落于以下图中的网格（坐标位置用一对数标识）里。



规划必须满足以下约束：

1. 汽车站(B)必须邻近道路
 2. 行政管理楼(A)和教室楼(C)必须邻近汽车站(B)
 3. 教室楼(C)必须邻近宿舍楼(D)
 4. 行政管理楼(A)一定不能靠近宿舍楼(D)
 5. 行政管理楼(A)一定不能建在一个山上
 6. 宿舍楼(D)一定要么建在一个山上，或者靠近道路
 7. 所有建筑必须在不同的网格里。
- 这里邻近的意思是指建筑所在网格间必须共享一条边，而不是一个角。

9-1) （3 分）指出在所有约束中（1-7），哪些属于一元约束？

9-2) （4 分）在应用所有一元约束后，请从下表中选择各变量的相应值域是什么？

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1)
<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)
<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)
<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)
<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)
<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)

9-3) (2 分) 从 9-2) 中的结果继续，现在应用我们讲过的弧一致性检查算法 (AC-3 算法)，初始时，弧队列里包含了所有的弧 (按照字母顺序排序好的)。让我们先来检查一下弧 $A \rightarrow B$ ，在之前 (9-2 里)，我们应用了一元约束，那时 A 和 B 的值域变成了以下：

A	B
(1,1)	
(1,3)	(1,3)
(2,2)	
(2,3)	(2,3)

那么现在，在我们检查了弧 $A \rightarrow B$ 的一致性以后，A 和 B 的值域又变成了什么？请从以下情况中选择。

A	B	A	B	A	B	A	B
(1,1)				(1,1)		(1,1)	
	(1,2)						
(1,3)	(1,3)	(1,3)	(1,3)	(1,3)		(1,3)	(1,3)
(2,2)		(2,2)		(2,2)		(2,2)	
(2,3)	(2,3)	(2,3)	(2,3)	(2,3)	(2,3)	(2,3)	
i		ii		iii		iv	

9-4) (4 分) 给出此时, 在强化检查完弧 $A \rightarrow B$ 的一致性后, 所有变量的值域 (9-3 中只选择了 A 和 B 的值域)? 请从以下列表中标识出来。

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1)
<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)
<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)
<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)
<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)
<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)

9-5) (4 分) 我们继续运行弧一致性检查的算法, 以上我们刚检查完弧 $A \rightarrow B$ 。接下来, 在对弧 $A \rightarrow C, A \rightarrow D, B \rightarrow A, B \rightarrow C, B \rightarrow D, C \rightarrow A$ 的一致性检查时, 各变量的值域实际没有被改变 (你可以自己验证)。那么, 现在请给出在接下来我们检查完弧 $C \rightarrow B$ 的一致性后, 各变量的值域包括哪些? 请从以下列表中标识出来。

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1)
<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)
<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)
<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)
<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)
<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)

9-6) (1 分) 此时, 在检查完弧 $C \rightarrow B$ 后, 因为 C 的值域变化了, 因此, 根据弧一致性检查算法 (AC-3 算法) 哪些弧被重新加入到弧队列中, 为了被重新检查它们的一致性? 记住, 此时队列中已有弧 $C \rightarrow D, D \rightarrow A, D \rightarrow B$, 和 $D \rightarrow C$ 。从以下中选择哪些弧被重新加入到弧队列中:

- ☐ $A \rightarrow B$
- ☐ $A \rightarrow C$
- ☐ $A \rightarrow D$
- ☐ $B \rightarrow A$
- ☐ $B \rightarrow C$
- ☐ $B \rightarrow D$
- ☐ $C \rightarrow A$
- ☐ $C \rightarrow B$
- ☐ $C \rightarrow D$
- ☐ $D \rightarrow A$
- ☐ $D \rightarrow B$
- ☐ $D \rightarrow C$
- ☐ None

9-7) (4 分) 接着上面的结果继续我们的弧一致性检查算法，记住，此时队列中包含弧 $C \rightarrow D$, $D \rightarrow A$, $D \rightarrow B$, $D \rightarrow C$ ，和前面我们在检查完弧 $C \rightarrow B$ 后所加入的弧。请从下面的列表中标识选择弧一致性检查算法结束后（即弧队列为空时），各变量的值域是什么？

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1)
<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)
<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)
<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)
<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)
<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)

9-8) (3 分) 此时我们已完成了弧一致性检查, 接着上面的结果, 我们发现此时, 各变量中的值域不是都是唯一的, 所以, 我们将开始对这个约束满足问题进行搜索(回溯搜索)以决定各变量的赋值。利用最小剩余值(MRV)的启发信息, 决定我们将从哪一个变量开始赋值? 从以下选择:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

9-9) (2 分) 你以上选择的变量此时应该只剩两个值。继续搜索之前, 我们将用最少约束值(LCV)启发信息来决定把哪一个值赋给该变量。选择最小约束值的方法是, 强化弧一致性的检查, 来看哪一个值, 使其他未赋值变量的值域里所剩值的数量之和最大, 就选这样一个值。从以下列表中选出要赋给(9-8 中选出的)该变量的值:

- ☐ (1,1)
- ☐ (1,2)
- ☐ (1,3)
- ☐ (2,1)
- ☐ (2,2)
- ☐ (2,3)

9-10) (4 分) 在上面赋完一个值以后, 在对下一个变量赋值前, 回溯算法会利用弧一致性检查来进行约束的传递, 以消减剩余未赋值变量的值域(相当于剪枝搜索空间)。

从下表中选择此时(在接着 9-8 和 9-9 的结果进行, 并运行完了弧检查算法以后)各变量的最终值域:

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1) ●	<input type="checkbox"/> (1,1)
<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)	<input type="checkbox"/> (1,2)
<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)	<input type="checkbox"/> (1,3)
<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)	<input type="checkbox"/> (2,1)
<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)	<input type="checkbox"/> (2,2)
<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)	<input type="checkbox"/> (2,3)

9-11) (1 分) 问此时的(9-10 中的)解是否是本约束满足问题的一个解?