

## 6.034 人工智能

期末考试

2000 年秋季

名字	
电子邮箱	
助教	
阅卷老师	

问题号	分值	得分	等级
问题 1	20		BDLPRUDL-PW
问题 2	19		BDLPRUDL-PW
问题 3	12		BDLPRUDL-PW
问题 4	18		BDLPRUDL-PW
问题 5	17		BDLPRUDL-PW
问题 6	14		BDLPRUDL-PW
总计	100		BDLPRUDL-PW

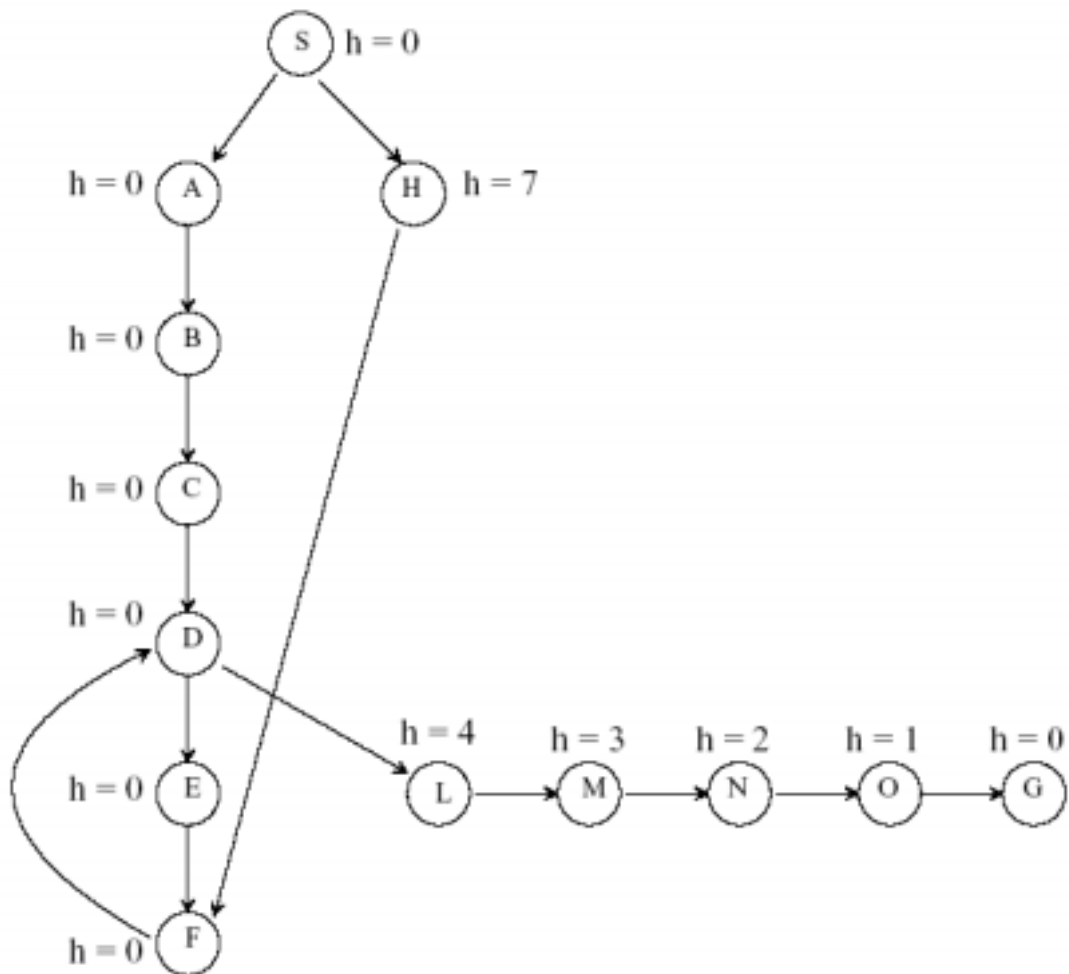
## 问题 1, 搜索 (19 分)

### 第 1 部分 (11 分)

这个问题和 1999 年期末的一个问题类似, 但是一些具体的重要信息不相同。在下面的有向图中你试着用不同的搜索算法寻找一条从状态 S 通往状态 G 的路径。

请不要使用访问过列表或者是扩展列表。

为方便起见, 在试卷的最后重印了该图, 它们可以撕下来。



先做如下假设:

- 所有连线长度都为 1。
- 从每个节点到目标节点的大致距离 ( $h$ ) 显示在每个节点的旁边, 这对你有启发作用。每个距离被认为是少于或者刚好等于真实距离。
- 任何一种搜索法则都不会产生循环路径。
- 深度优先和宽度优先的搜索法则都是以字母顺序对分支进行遍历。

- 和习题集一样，搜索算法使用一个搜索队列。
- 请不要使用访问过的列表或者是扩展列表。

按指定的搜索方法记录下扩展节点的顺序。

注意：在搜索中要求你去完成任务，中间节点在节点序列中有可能出现不止一次。

写出深度优先遍历的节点次序（我们已为你开了个头）：

S-A-B-

写出  $A^*$  顺序（我们已为你开了个头）。在你遍历顺序的节点上方写出号码， $A^*$  用它来依次表示接下来要扩展的节点。提示：在一张废纸上作出该搜索树。记住，

不要使用访问过的或者是扩展列表（所以不要去掉通往中间节点的多余路径），但是你可以使用对你有启发的剩余距离的估计值。（这

是对带有启发式剩余距离估计的分支限界方法的另外一种说法。）

0 1 2  
S-A-B-

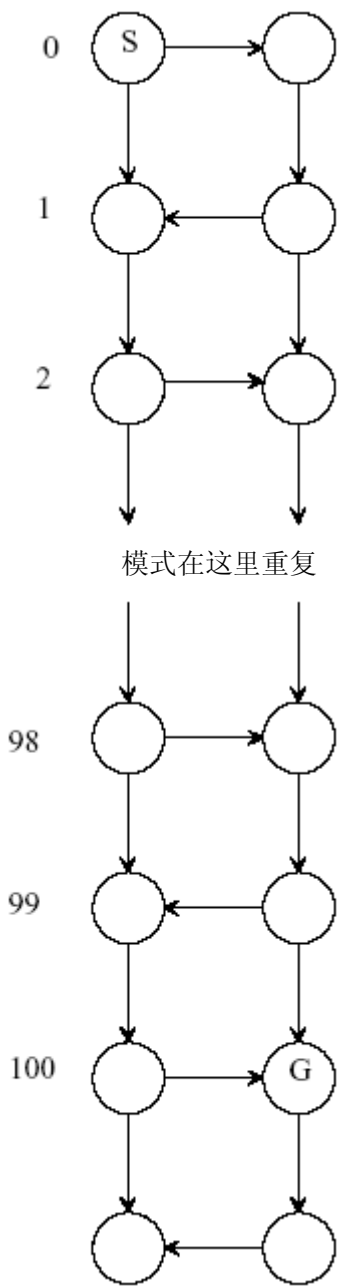
对  $A^*$ ，指出最后找到的路径...

S-

...还有它的长度

## 第 2 部分（8 分）

考虑下面的有向图，其中所有连接长度都是 1。



假设你采用英国博物馆搜索算法进行搜索，不要使用任何机制去消除通往中间节点的多余路径，找到目标节点 G 需要多少个节点？提示：作出部分搜索树。

假设是带有动态规划的  $A^*$  搜索，使用剩余路径长度为 0 的启发式估计，有多少条路径能延伸到目标节点，G。

## 问题 2，规则（20 分）

注意这个问题使用了与在线问题集同样的一些基本概念，但是在重要的地方是不同的。

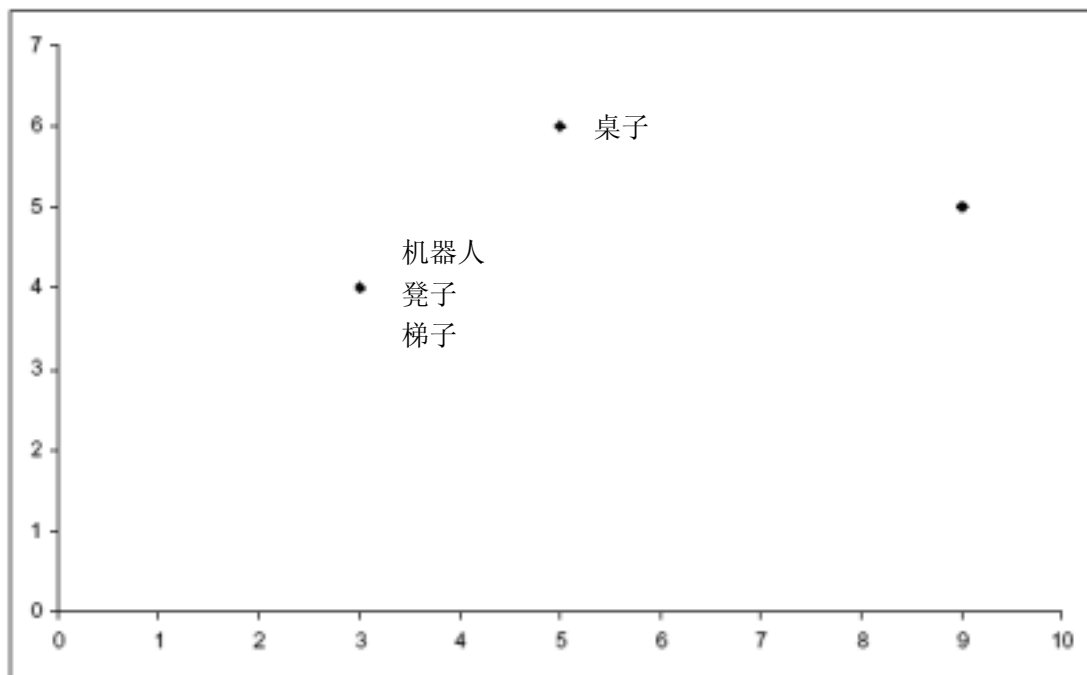
假设我们有一个机器人在一个有着很多其他物体的房间里可以到处移动，每个物体和机器人都有由一对数（格子坐标中的 x 和 y 坐标）确定的位置。机器人也能爬到物体上。我们用如下类型的断言来描述世界的状态：

(机器人 在 <位置>) ; 表明机器人的位置  
; 注意这和问题集中的不同  
(机器人 在...之上 <物体>) ; 也和问题集中的不同  
(<物体 在 <位置>)  
(位置 <位置>)

(机器人 在 <位置>) 和 (机器人 在...之上 <物体>) 断言描述机器人的状态。( <物体 在 <位置>) 断言描述每个物体的位置。有个特殊的物体，地板，没有指定的位置。一个位置的存在通过 (位置 <位置>) 来表示。机器人能够完成两个任务。

- **行走**—机器人行走的前提条件必须是在地板上，然后它将在新的位置上。
- **攀登**—有两种情况。从一个物体上往下爬：机器人可以在还没有到达地板上的任何时候往下爬。爬到物体上去：机器人必须在地板上和物体处于同一坐标。爬上去之后，机器人在相应的物体上面。

样本安排：



我们可以想象使用一些规则去模拟这种情形。考虑到上面指出的表示法的不同点，将这些规则在很小但是很重要的方面设置得和那些在线问题不同。

```
(climb-down
  IF
  (robot on ?support)
  AND-IF (not (equal? '?support 'floor))
  SAYING "CLIMB down to floor"
  ADD (robot on floor)
  DELETE (robot on ?support))

(walk
  IF
  (robot on floor)
  (robot at ?old-loc)
  (loc ?new-loc)
  AND-IF (not (equal? '?old-loc '?new-loc))
  SAYING "WALK to " ?new-loc
  ADD (robot at ?new-loc)
  DELETE (robot at ?old-loc))

(end ;-- rule just terminates chaining when the goal is accomplished
  IF
  (goal ?g)
  ?g
  SAYING "Stop"
  ADD (stop))
```

这是上页用坐标图所描绘的例子的一些断言。

```
(loc (9 5))
(loc (3 4))
(loc (5 6))
(机器人在 (3 4))
(机器人在...之上 地板)
(桌子在 (5 6))
(凳子 在 (3 4))
(梯子 在 (3 4))
(目标 (在...之上 桌子))
```

### 第 1 部分 (10 分)

A) (6 分) 指出前向推理链的三种循环的结果，假设

- 你按照上面断言所描述的精确位置开始，
- 你可以使用上面所给出的三种规则（停止，向下爬，步行），
- 系统所使用的前向推理链工作方式和问题集所用的一样，而且以来规则顺序解决冲突，
- 新的断言要加到断言列表的最上面。

在每次循环中指出实施了哪条规则，在规则实施之后罗列出应该出现的断言。（为了给你节

省一些工作，我们已经写出了开始的断言列表。编辑这些断言指出在每个规则实施之后哪些断言会相应出现。)

实施的规则 (规则名和变量约束)	在规则实施之后的断言设置。 划出没有出现的，添加新的断言
	(loc (9 5)) (loc (3 4)) (loc (5 6)) (机器人 在 (3 4)) (机器人 在...之上 地板) (桌子 在 (5 6)) (凳子 在 (3 4)) (梯子 在 (3 4)) (目标 (在...之上 桌子))
	(loc (9 5)) (loc (3 4)) (loc (5 6)) (机器人在 (3 4)) (机器人 在...之上 地板) (桌子 在 (5 6)) (凳子 在 (3 4)) (梯子 在 (3 4)) (目标 (在...之上 桌子))
	(loc (9 5)) (loc (3 4)) (loc (5 6)) (机器人 在 (3 4)) (机器人 在...之上 地板) (桌子 在 (5 6)) (凳子 在 (3 4)) (梯子 在 (3 4)) (目标 (在...之上 桌子))

B) (4分) 问题集中要求你写一个向上爬的规则。我们现在给出一个并要求你使用它。规则是：

```
(climb-up
  IF
    (robot at ?location)
    (?support at ?location)
  SAYING "CLIMB onto " ?support
  ADD      (robot on ?support)
  DELETE  (robot on floor))
```

把这条规则加到规则集的最前面，考虑一下用初始断言集进行正向推理时发生的情况。你应该在向上爬的规则中发现两个错误（我们在下面重印了这些断言）。接下来，修改它，并写一个更好版本（又为了节约你的时间，我们在下面重印了这些断言以便你编辑它们的时候更加方便。）初始断言如下：

```
(loc (9 5))
(loc (3 4))
(loc (5 6))
(机器人 at (3 4))
(机器人 on 地板)
(桌子 at (5 6))
(凳子 at (3 4))
(梯子 at (3 4))
(目标 (on 桌子))
```

你修改过后的规则：

```
(climb-up
  IF
    (robot at ?location)
    (?support at ?location)
  SAYING "CLIMB onto " ?support
  ADD (robot on ?support)
  DELETE (robot on floor))
```

## 第 2 部分（10 分）

下面给出了一组不带变量的简单推导规则。用这些规则从目标（Z）进行反向推理。

为了简化后向推理链，假设数据库中开始没有信息而且使用者对每个问题的回答都是“是”。

例如，我们只有一个规则

```
如果 (P)
      (Q)
那么 (R)
```

系统开始应当向使用者提问“P? ”，使用者要回答“是的”等等。

假设系统以下面列出的顺序尝试每个规则。

R1

如果 (A)  
      (B)  
那么 (Z)

R2

如果 (C)  
      (D)  
那么 (Z)

R3

如果 (E)  
那么 (Z)

R4

如果 (F)  
那么 (D)

R5

如果 (G)  
那么 (E)

R3

如果 (H)  
那么 (F)

R7

如果 (J)  
那么 (A)

R8

如果 (K)  
那么 (J)

A) (2分) 指明所问问题的顺序并把它们的字母代号列入下表。一般我们会给你足够的空格。

1.
2.
3.
4.

5.

B) (2分) 现在画出完整的搜索树，从 (Z) 开始后向推理链，假设对所有问题进行否定回答：



C) (2分) Ben Bitdiddle 指出我们系统的行为是可以改善的。不采用传统用于反向推理链的深度优先规则，Ben 提出了一种你可以称它为“最短剩余路径”的规则。系统看起来更加灵巧因为它尝试的是最短路径，例如，更短的推理链。

为了实现它，Ben 将从一棵树的叶节点到目标节点的所有路径全部罗列出来。在上面 B 部分已经显示了所生成的树的路径列表（你有足够的空格；我们已向你提供了一条路径以说明我们的意思和所使用的符号）：

1. B Z
2.
3.
4.
5.
6.
7.
8.

D) (2分) Ben 为了实现他的想法，然后就设计了一个简单的评分功能：每条路径的分数就是它的长度，低分比高分更有优先权（通过使用规则顺序打破原有关系）。使用 Ben 的

评分功能去决定搜索树的搜索顺序，在下面的表格中列出所问问题的顺序。和上面一样假设所有问题都是否定回答。

1.
2.
3.
4.
5.

E) (2 分) 在试验完 Ben 的建议之后你会认识到一些事情。选择一个能最好表达 Ben 修改版本的特点的叙述。

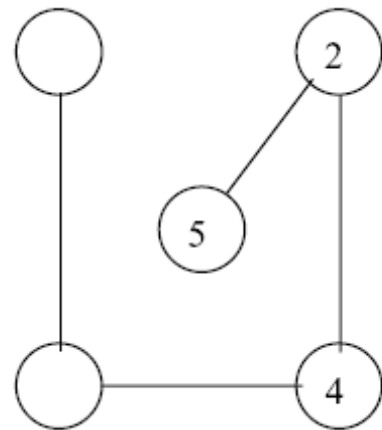
- 总是比前面的方法所问问题要多。
- 总是比前面方法所问问题要少。
- 所问问题总是和普通深度优先规则的顺序不相同。
- 所问问题从没有和普通深度优先规则的顺序不同。
- 可能比普通深度优先规则要创建更大的推导树。
- 可能比普通深度优先规则要创建更小的推导树。
- 以上都不正确。

### 问题 3，约束满足（12 分）

注意—这个问题的几个部分可以独立完成。

考虑下面的约束图。假设每个变量有相同的取值范围  $D_i = \{A, B, C\}$ 。每对约束变量的唯一正确的赋值如下表所示。

约束条件 ( $V_i-V_j$ )	正确赋值 ( $V_i-V_j$ )	
1-3	A-C	B-A
2-4	A-A	B-B
3-4	A-B	C-A
2-5	B-A	A-C



#### 第 1 部分（7 分）

A. 重复进行约束传播直到弧相容，然后写下约束传播之后每个变量的合理值。

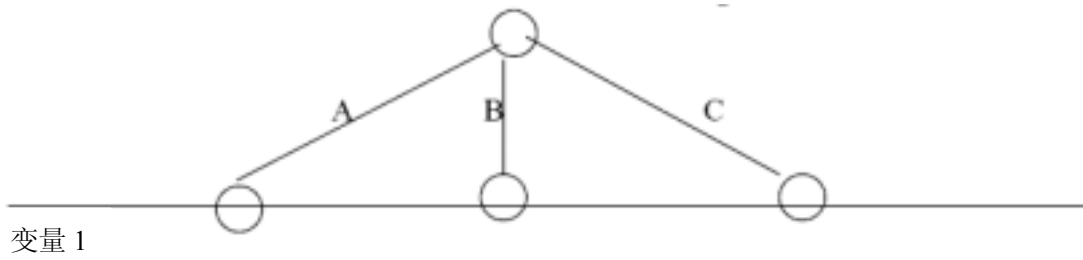
变量	合理值
1	
2	
3	
4	
5	

B. 如果只是在 A 部分给出合理值的列表，你能说出这个问题可能解的数目吗？可以给出一个明确的数目或者是一个明确的范围（指出范围的最大最小值）。

#### 第 2 部分（5 分）

以下，没有作任何约束限制。从头开始！

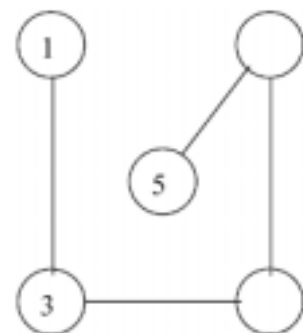
使用超前检查的回溯法 (BT-FC) 找到这个问题的一个正确结果。分别按数字顺序检查变量，按字母顺序检查结果数值。画出下面的搜索树，我们已经给你开了个头。对于树中的每个节点只画出相应正确的子节点。每条水平线表示与指定变量值的赋值相符合的树的等级。按考虑过的赋值顺序对树中的节点进行编号。



找到的答案是：

<b>1</b>	
<b>2</b>	
<b>3</b>	
<b>4</b>	
<b>5</b>	

约束条件 ( $V_i-V_j$ )	正确分配 ( $V_i-V_j$ )	
<b>1-3</b>	<b>A-C</b>	<b>B-A</b>
<b>2-4</b>	<b>A-A</b>	<b>B-B</b>
<b>3-4</b>	<b>A-B</b>	<b>C-A</b>
<b>2-5</b>	<b>B-A</b>	<b>A-C</b>

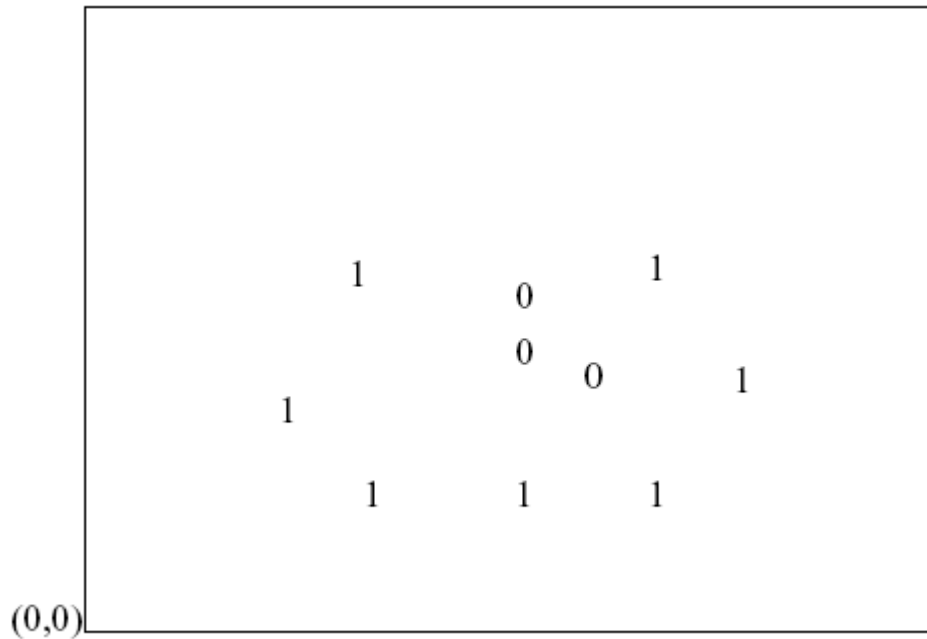


为了方便已给你复制了该表格

## 问题 4, 机器学习 I (18 分)

注意—这个问题的几个部分可以独立完成。

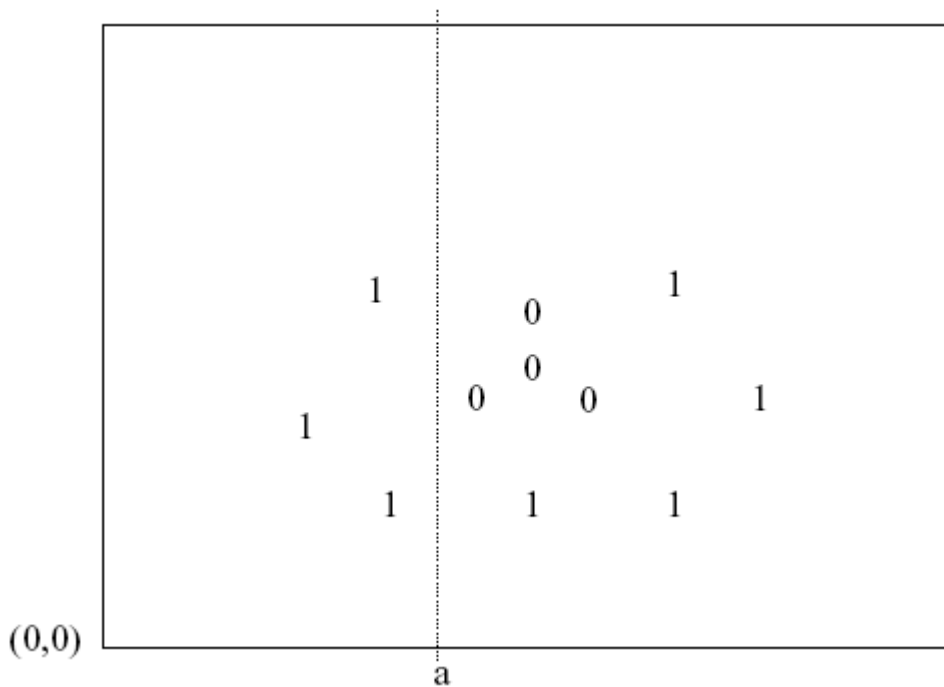
### 第 1 部分 (4 分)



- 假设你决定 1 最近邻的规则来辨认。选出最小的一组点（从上面给出的点中），如果这些点按照 1 最近邻的规则来编排，请给出所有点的分类。圈出所选择的点。
- 如果你选择的点只是训练集内的点，那么画出产生的决策边界。

### 第 2 部分 (6 分)

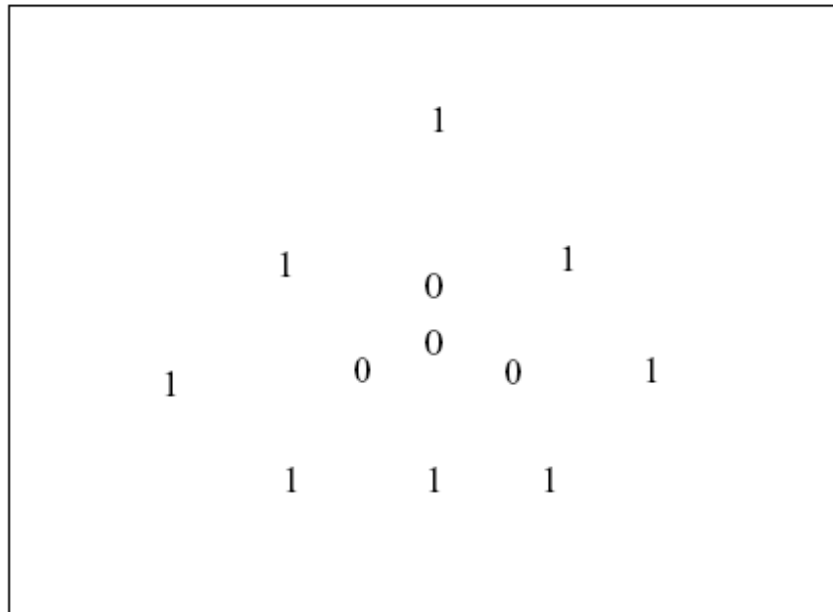
注意这个数据组和**第一部分**的不相同。



- A. 上面给出的测试的平均无序的值是多少？你可以以分数形式和未化简形式给出结果，但我们希望你尽可能简化它（不要用计算器）。

- B. （在上图中）和 A 部分一样，画出用于对数据进行分类的完整的 ID 树的决策边界。选择贪婪算法所采用的测试方法（但不用完成所有的运算）。用字母在图中每个测试的特殊值上做个记号（图中我们已经按此方法做了记号 a）。
- C. 在下面画出完整的树（确保能够指定测试的哪些分支是肯定的，哪些分支是否定的，还要能确保输出为分类）。使用  $X < a$  形式的测试。

### 第 3 部分 (8 分)



(0,0)

- A. 画出最简单的神经网络（神经元和连线），该结构应该能够把上面的数据组进行分类。包括每个单元的偏移量/阈值。假设每个单元都是感知器。用数字标记每个单元，并作为下一个问题的参考。



- B. 你结构的权值是多少（包括偏移量/阈值权值）？

- C. 在上面的数据组中为 A 部分的解答中的每个单元画出决策边界，A 部分的输入为 X 和 Y。

用单位数字来标记每个分界线。标出分界线哪边是 0 哪边是 1。

- D. 挑选出一个单元，其决策边界你已在 C 中画出，并给出这个单元权值。假设 X-Y 特征空间的起始位置在图的左下角，尺寸是 1。注意，对你所选的单元， $W_x$  表示 X 方向的权值， $W_y$  表示 Y 方向的权值， $W_o$  表示偏移量/门槛值的权值。

选择的单元：



权值：

$W_x =$
$W_y =$
$W_o =$

## 问题 5, 机器学习 II (17 分)

### 第 1 部分 (5 分)

请简要回答。

- A. 为了使用 SVM 对问题 4 中第 1 部分进行分类, 你应该选择哪一类核函数?

- B. 假设选择了一个足够小的步长而且等了足够长时间, 反向传播保证能找到误差的全局最小值。正确还是错误?


- C. 为什么我们要把一个 S 神经网络的初始权值设为一个小数字而不是一个大数字?

- D. 在带有 N 个分类和 M 个例子的任意一个数组的测试中, 平均无序度的最大值是多少?

- E. 你如何验证某特征是否对采用 1 最近邻法的分类器进行正确分类有所帮助呢?

**第 2 部分 (6 分)**

- A. 假设用最近邻, (简要) 描述规格化数据对分类的准确性有损害的情形。或者说明为什么这种情形不可能发生。



- B. 假设用判定树, (简要) 描述规格化数据对分类的准确性有损害的情形。或者说明为什么这种情形不可能发生。



### 第3部分 (6分)

选择任何一种**通常**与指定学习方法中过学习相关联的行为/情形。每种学习方法你可以选择一个或多个项目。

#### A. K 的最近邻

- 增加了 K 的尺寸
- 减少了 K 的尺寸
- 以上都不正确

#### B. 判定树

在这里，你假设所有标本在叶节点被分割成相同的组之前判定树的使用者已经停止了树的生长。

- 增加了树的深度
- 减少了树的深度
- 增加了一个叶节点上的标本数（通过减少深度）
- 减少了一个叶节点上的标本数（通过增加深度）
- 以上都不正确

#### C. 神经网络

- 增加了系统的单元数（神经元）
- 减少了系统的单元数（神经元）
- 增加了训练循环的次数
- 减少了训练循环的次数
- 以上都不正确

#### D. SVM

现在你要仔细研究下面的径向基函数核。

- 当使用径向基函数核时减少 sigma
- 当使用径向基函数核时增加 sigma
- 以上都不正确

$$K(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}}$$

## 问题 6, 其他 (14 分)

许多问题和第二次测验中的问题类似。但是都不完全相同。

选择一个答案以完成以下题目, 多选无效, 每个问题一分, 选错不扣分:

逐步加深或叫做反复加深在博弈中能够表现出色是因为:

- $\alpha - \beta$  允许你在典型博弈树中搜索接近两倍的深度。
- 层与层之间的分支因数不同。
- 在给定深度的博弈树中几乎没有哪个节点是在最后一层上。
- 一层中的节点数目随着深度成指数增加。
- 以上都正确。
- 以上都不正确。

$\alpha - \beta$ :

- 是最小最大法速度的两倍。
- 和最小最大法速度相等。
- 当时间用完时保证会有答案。
- 在典型博弈树中搜索深度接近最小最大法速度的两倍。
- 以上都正确。
- 以上都不正确。

一个框架系统可能有用是因为它提供了:

- 特性继承。
- 方法继承。
- 特定域的词汇。
- 特定域的数值。
- 以上都正确。
- 以上都不正确。

空间转换表示法的主要优点在于它:

- 集中于变化。
- 能够被转换成关系数据库的记录。
- 为语义转换树的语法提供了一个基础。
- 确保所有描述都满足牛顿力学。
- 以上都正确。
- 以上都不正确。

利用 Ullman 的队列方法（预知点位置）进行辨识要求具有：

- 找到相应点或成组的点的能力。
- 刚性物体（例如，不能是松软的物体，就像一套衣服那样）。
- 库中包含了每个需要辨识的物体的多个图像。
- 求解线性方程的一种机制。
- 以上都正确。
- 以上都不正确。

三个图像（模型）中的四个点足够预知第四个图像中的点的位置除非：

- 在所有的模型图像中并非所有点都能看见。
- 模型图像中的点并非正确对应。
- 三个模型图像中的不同点都是由一个单轴旋转产生的。
- 三个模型图像中的不同点都是由一个单轴平移产生的。
- 以上都正确。
- 以上都不正确。

用于学习音位规则的 Yip-Sussman 方法：

- 一般化种子模式。
- 特殊化种子模式。
- 特殊化和一般化种子模式。
- 使用 A\* 搜索找到最小模式。
- 以上都正确。
- 以上都不正确。

Yip-Sussman 方法成功的一个原因是：

- 提供了数千个例子。
- 学习的规则是双向的。
- 包含了稀疏的高维空间。
- 包含了二元空间。
- 以上都正确。
- 以上都不正确。

从标本中进行似是而非学习成功的一个主要原因是：

- 提供了数千个例子。
- 老师忽视了学生知识程度的状态。
- 学生被要求只是专门研究成果例子。
- 学生自己产生似是而非的例子。
- 以上都正确。
- 以上都不正确。

似是而非学习中：

- 似是而非（还没成为实例），导致一般化（并非特殊化）。
- 禁止连接的试探法是由实例而不是似是而非触发的。
- 停止连接的试探法是由实例而不是似是而非触发的。
- 点连接的试探法是由实例而不是似是而非触发的。
- 要求连接的试探法是由实例而不是似是而非触发的。
- 以上都正确。
- 以上都不正确。

利用解释的学习：

- 从特殊情形中归纳。
- 加速后来的推理。
- 能够影响实际中的推论。
- 依靠适当的表示法。
- 以上都正确。
- 以上都不正确。

好的表示法：

- 使得正确的细节清晰。
- 抑制不相关的细节。
- 揭示约束。
- 把有帮助的细节聚合到一起。
- 以上都正确。
- 以上都不正确。

一只老鼠的海马状突起：

- 似乎在睡觉的时候也能进行一种活动。
- 似乎与形成长期记忆有关。
- 似乎有神经元对老鼠的位置作出反应。
- 似乎支持展示归纳能力的机制。
- 以上都正确。
- 以上都不正确。

如果你在高校失去了你的脑部活动，那么你将：

- 不能通过 6.034 考试。
- 不能认出你的朋友。
- 不能走路。
- 不能说话。

- 以上都正确。
- 以上都不正确。

为了方便给你重复提供图表。

