

第三章 搜索与求解

1. 以下关于用搜索算法求解最短路径问题的说法中，不正确的是（ ）。
- A. 给定两个状态，可能不存在两个状态之间的路径；也可能存在两个状态之间的路径，但不存在最短路径（如考虑存在负值的回路情况）。
 - B. 假设状态数量有限，当所有单步代价都相同且大于 0 时，深度优先的图搜索是最优的。
 - C. 假设状态数量有限，当所有单步代价都相同且大于 0 时，广度优先的图搜索是最优的。
 - D. 图搜索算法通常比树搜索算法的时间效率更高。

答案：B

A 如果图不连通，则可能不存在路径。如果图中存在负值回路（当然还有其他情况），则可能不存在最短路径。

B 显然不是最优的。

C 在这种情况下，节点所在层数和其路径长度是成正比的，因此优先扩展浅层节点等价于优先扩展路径代价小的节点，这在图搜索中是最优的（可参见 Dijkstra 算法）。

D 因为图搜索是在树搜索的基础上进一步剪枝，因此扩展的节点数量通常更少。

2. 以下关于启发函数和评价函数的说法中正确的是（ ）。
- A. 启发函数不会过高估计从当前节点到目标结点之间的实际代价。
 - B. 取值恒为 0 的启发函数必然是可容的。
 - C. 评价函数通常是对当前节点到目标节点距离的估计。
 - D. 如果启发函数满足可容性，那么在树搜索 A*算法中节点的评价函数值按照扩展顺序单调非减；启发函数满足一致性时图搜索 A*算法也满足该性质。

答案：D

A 只有可容的启发函数才不会过高估计从当前节点到目标结点之间的实际代价。

B 如果存在负值边，则很容易构造反例。

C 启发函数通常是对当前节点到目标节点距离的估计，评价函数不一定有实际意义。

D 根据对 A*算法的分析，不难证明。

3. 假如可以对围棋的规则做出如下修改，其中哪个修改方案不影响使用本章介绍的 Minimax 算法求解该问题？（ ）
- A. 由双方轮流落子，改为黑方连落两子后白方落一子。
 - B. 双方互相不知道对方落子的位置。
 - C. 由两人对弈改为三人对弈。
 - D. 终局时黑方所占的每目（即每个交叉点）计 1 分，且事先给定了白方在棋盘上每个位置取得一目所获取的分数，假设这些分数各不相同。双方都以取得最高得分为目标。

答案：A

A 只需要重新定义黑方的动作为每次落两子即可。

B 导致问题中信息不完全，因此 Minimax 算法无法求解。

C 导致问题不再是两人对抗问题，每个人的目标不能再简单地用最大化/最小化某一个人的分数来衡量。

D 使该问题不是零和博弈。白方最大化自己的分数不一定必须最小化黑方的分数。

4. 下列关于探索与利用的说法中，不正确的是（ ）。
- A. 在多臂赌博机问题中，过度探索会导致算法很少主动去选择比较好的摇臂。
 - B. 在多臂赌博机问题中，过度利用可能导致算法对部分臂膀额奖励期望估计不准确。
 - C. 在 ϵ 贪心算法中， ϵ 的值越大，表示算法越倾向于探索。
 - D. 在多臂赌博机问题中，某时刻 UCB1 算法选择的臂膀置信上界为 R ，则此时任意摇动一个臂膀，得到的硬币数量不会超过 R 。

答案：D

A、B、C 显然正确。

D 中置信上界的含义是样本取值以极大的概率不会超过置信上界，并不是说不可能超过。

5. 下列关于蒙特卡洛树搜索算法的说法中，不正确的是（ ）。

- A. 选择过程体现了探索与利用的平衡。
- B. 算法进入扩展步骤时，当前节点的所有子节点必然都未被扩展。
- C. 模拟步骤采取的策略与选择步骤不一定要相同。
- D. 反向传播只需要更新当前路径上已被扩展的节点。

答案：B

A 选择过程中 UCB1 算法即体现了探索与利用的平衡。

B 只要有一个子节点未被扩展，算法就会进入扩展步骤。

C 模拟步骤的策略不一定要和选择步骤相同，模拟步骤通常会采取更简单的策略。

D 对。更新当前路径上的节点，且不在搜索树中的当然不用更新。

6. 如图 1 所示，假设每个节点代表一个状态，节点之间的箭头表示状态转移关系，箭头旁的数字表示状态转移的代价。若使用以下搜索算法寻找从状态 A 到状态 I 的路径，请画出算法终止（找到第一条路径）时的搜索树，并在搜索树中标出节点的扩展顺序，以及找到的路径。若有多个节点拥有相同的扩展优先度，则优先扩展对应路径字典序较小的节点。

1) 基于树搜索的广度优先搜索。

2) 基于图搜索的深度优先搜索。

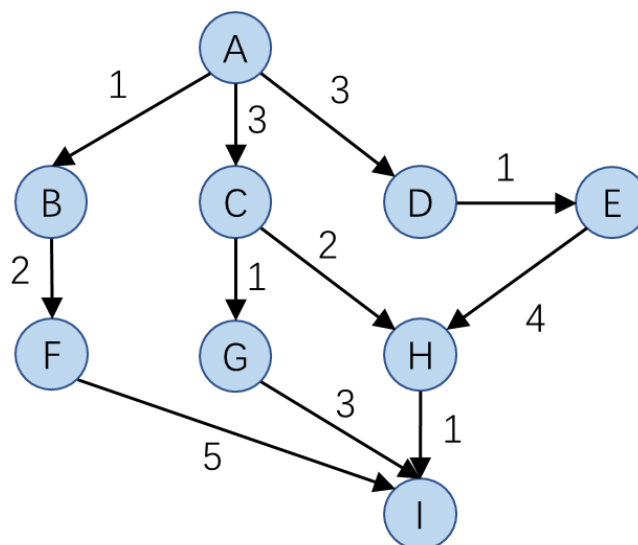
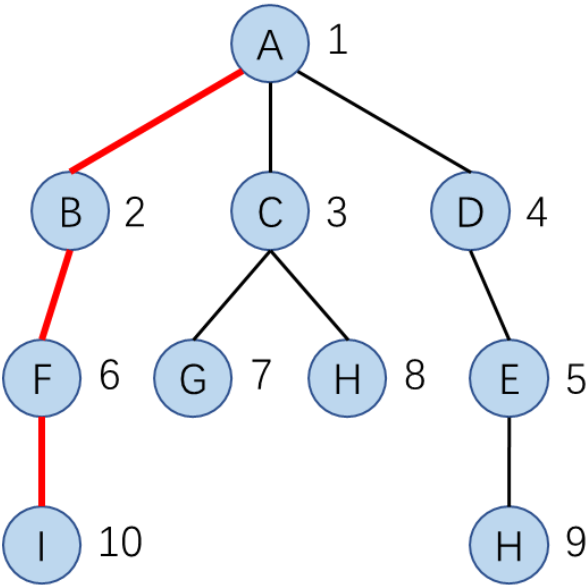


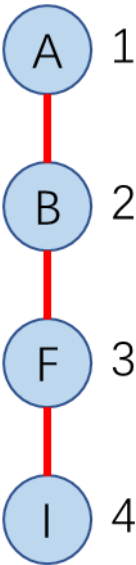
图 1 状态转移图

答案：下图中红线表示路径，节点旁的数字表示扩展顺序。

(1)



(2)



7. 考虑图 1 中的问题，给定每个状态的启发函数如表 1 所示。若仍以状态 A 为初始状态、状态 I 为终止状态，请分别使用以下算法求解从 A 到 I 的路径，并按照第 6 题中的方法画出搜索树。若有多个节点拥有相同的扩展优先度，则优先扩展对应路径字典序较小的节点。

表 1 启发函数的取值。

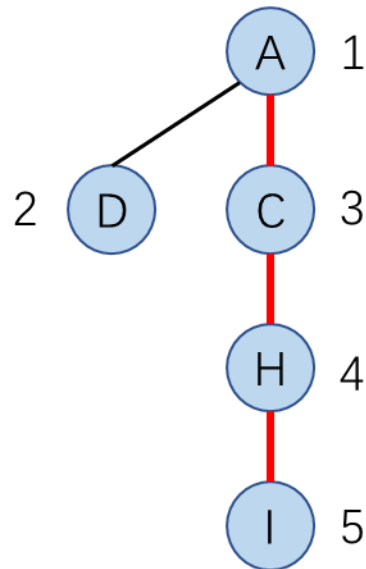
状态	A	B	C	D	E	F	G	H	I
启发函数	5	4	3	2	5	5	2	1	0

(1) 基于树搜索的贪婪最佳优先搜索。

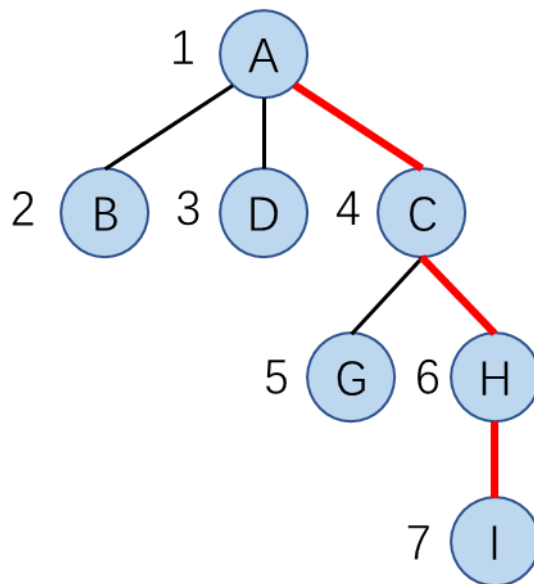
(2) 基于图搜索的 A* 算法。

答案：下图中红线表示路径，节点旁的数字表示扩展顺序。

(1)



(2)



8. 根据问题 7 中的搜索过程，回答下列问题。

- (1) 贪婪最佳优先搜索和 A* 算法均能找到最短路径，且贪婪最佳优先搜索扩展的节点数更少，这是否与正文中的结论“A* 搜索是在已知信息下同类搜索策略中最优的”相矛盾，为什么？
- (2) 不难验证，问题 7 中给出的启发函数既是可容的也是一致的，从这个角度来说这是一个“好”的启发函数。显然这个启发函数并不足够“好”，以至于 A* 算法在该问题中效率低于贪婪最佳优先搜索。那么怎样的启发函数才能提高搜索效率？搜索效率最高时启发函数应该如何取值？

答案:

- (1) 不矛盾。贪婪最佳优先搜索并不具有最优性，本题中的例子只是它能找到最优解的一个特例。“A*搜索是在已知信息下同类搜索策略中最优的”，其含义是：额外信息仅包括当前的启发函数时，所有能够保证最优性（即在任意有最短路径的问题中都能找到最短路径）的算法中，A*算法扩展的节点数量是最少的。
- (2) 启发函数在满足可容性或一致性的基础上，其值越接近当前节点到终止节点的最小代价，搜索的效率越高。当启发函数值等于当前节点到终止节点的最小代价时，算法每一步都会朝着最优的方向探索，以 $O(m)$ 的复杂度得到最优解。

9. 图 2 展示了一棵 Minimax 搜索树，可采用 alpha-beta 剪枝算法进行对抗搜索。假设对于每个节点的后继节点，算法按照从左向右的方向扩展。同时假设当 alpha 值等于 beta 值时，算法不进行剪枝。请问：

- (1) 对图 2(a)中所示搜索树进行搜索，请画出在算法结束时搜索树的状态，用“x”符号标出被剪枝的子树，并计算该算法扩展的节点数量。
- (2) 图 2(b)展示了和图 2(a)中完全相同的搜索树，只不过对搜索算法的顺序做了一些调整（扩展顺序交换的节点用红色虚线框标出）。请画出在算法结束时搜索树的状态，用“x”符号标出被剪枝的子树，并计算该算法扩展的节点数量。

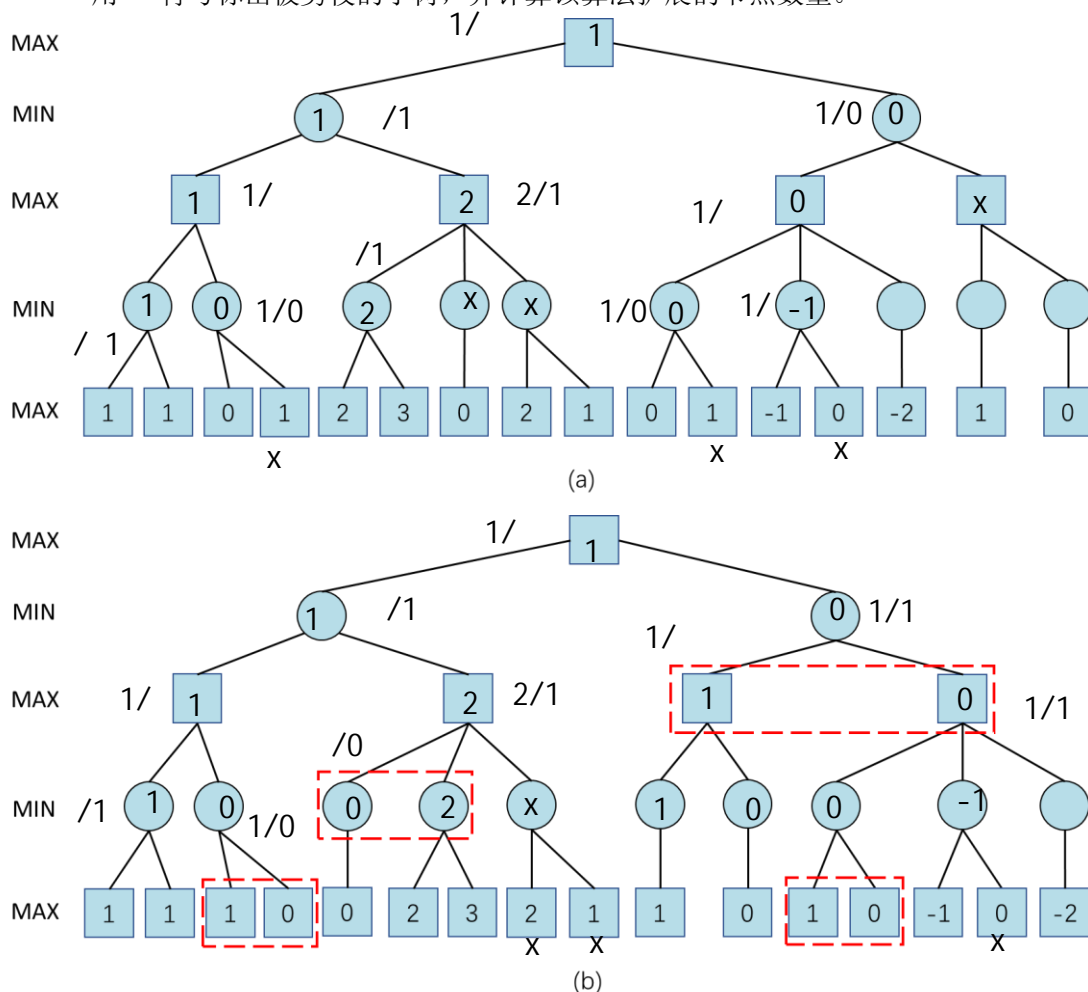
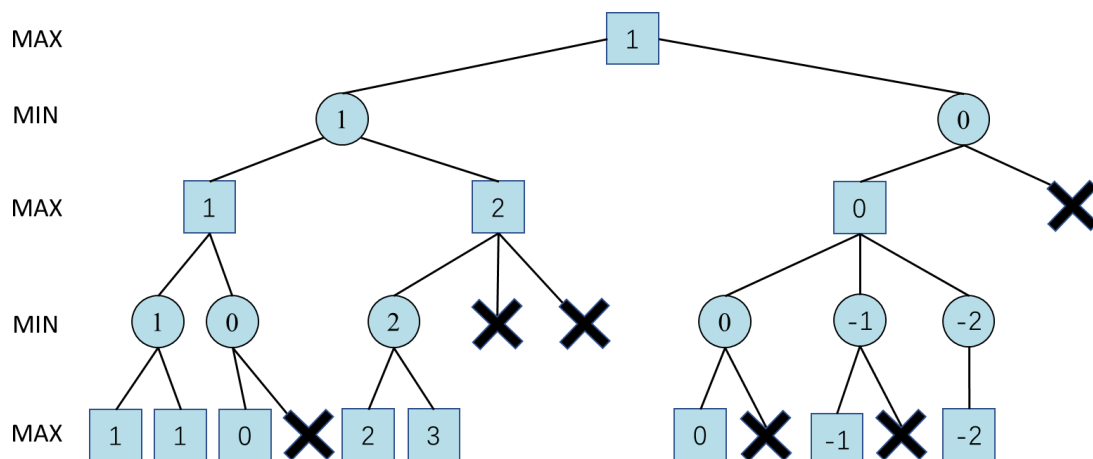


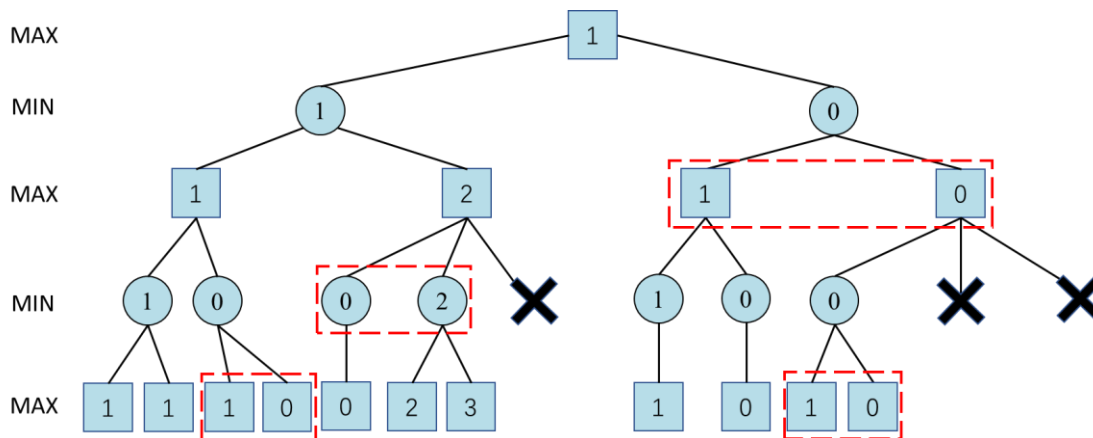
图 2 一棵 Minimax 搜索树。

1. 答案: 扩展情况如下。显然扩展顺序会影响最终扩展的节点数量（算法时间效率）。

(1) 扩展节点数量为 20。



(2) 扩展节点数量为 25。



10. 图 3 展示了一个蒙特卡洛树搜索的例子。其中每个叶子节点（终止节点）下标出了该节点对应的奖励。为了最大化取得的奖励，可利用蒙特卡洛树搜索求解奖励最大的路径。假设执行了若干步骤后，算法的状态如图 3 所示，节点内的数字分别表示“总奖励/访问次数”，虚线节点表示尚未扩展的节点。算法此时正要开始下一轮选择-扩展-模拟-反向传播的迭代

- (1) 假设 UCB1 算法中的超参数 $C = 1$ ，请计算并画出算法选择过程经过的路径。
- (2) 请继续执行扩展、模拟、反向传播步骤，并画出完成后的搜索树状态。（为了避免随机性，假设扩展总是扩展最左侧的未扩展节点，模拟总是选择最左侧的路径。）
- (3) 尝试进行若干次迭代，请问此时算法是否能有效地找到奖励最大的叶子结点（奖励为 9），那么进行足够多次迭代以后又如何？如果希望提高算法的效率，应该做出怎样的调整？

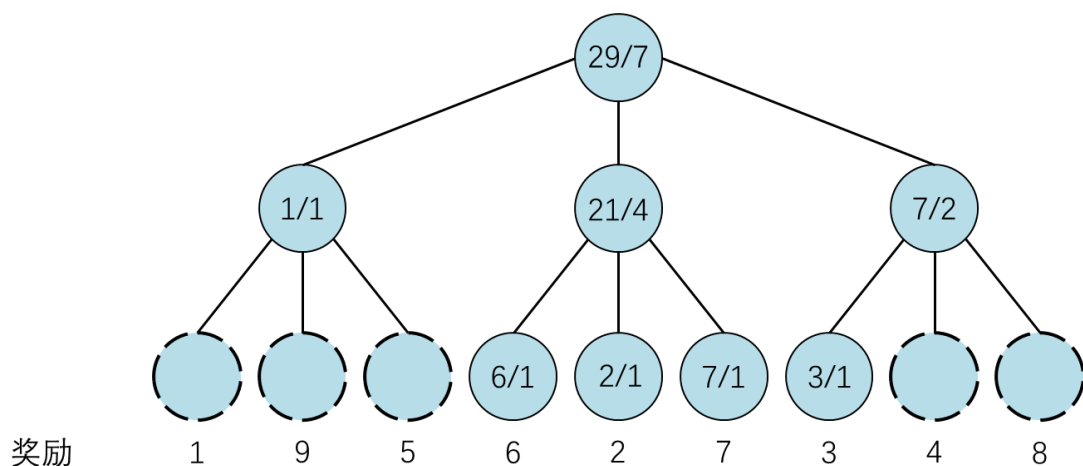
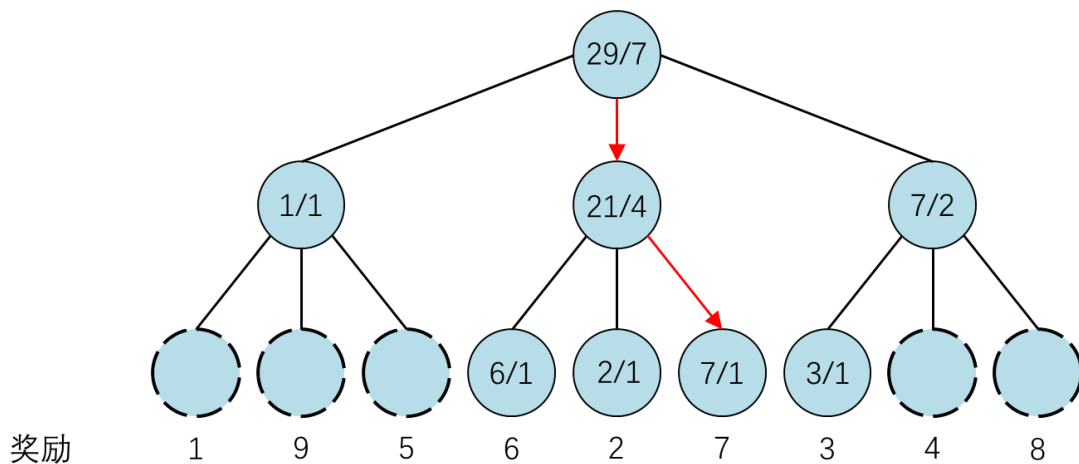


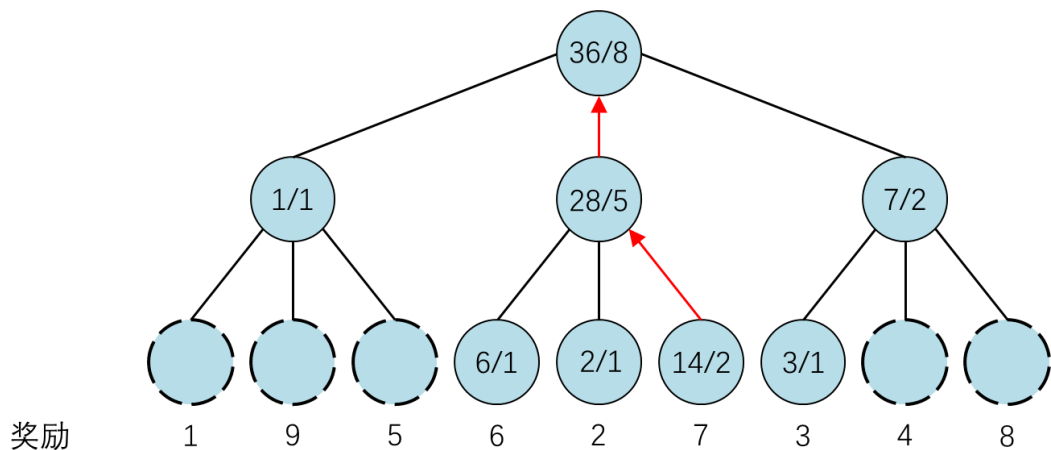
图 3 一棵蒙特卡洛树搜索的搜索树。

答案:

- (1) 第一步三个节点的 UCB 值从左到右分别为 $\frac{1}{1} + \sqrt{\frac{2\ln 7}{1}} = 2.97$, $\frac{21}{4} + \sqrt{\frac{2\ln 7}{4}} = 6.24$, $\frac{7}{2} + \sqrt{\frac{2\ln 7}{2}} = 4.89$, 因此第一步选择第二层中间的节点。第二步三个节点的 UCB 值从左到右分别为 $\frac{6}{1} + \sqrt{\frac{2\ln 4}{1}} = 7.67$, $\frac{2}{1} + \sqrt{\frac{2\ln 4}{1}} = 3.67$, $\frac{7}{1} + \sqrt{\frac{2\ln 4}{1}} = 8.67$, 因此第二步选择奖励为 7 的节点。如下图所示。



- (2) 由于此时已经到达叶子节点，因此不需要进行扩展和模拟过程，反向传播后结果如下图所示



- (3) 算法在很长一段时间内都会选择奖励为 7 的节点，而不会探索奖励为 9 的节点。当实验次数足够多时，第二层左侧的节点的 UCB 值最终会超过第二层中间节点的 UCB 值，因此只要实验次数足够多，算法是有可能探索到奖励为 9 的节点的。如果希望提高算法的效率，可考虑加大探索的力度，即取一个更大的超参数 C 。不难验证，在原题中的状态下，取 $C = 10$ 即可令算法选择第二层左侧的节点。

第四章 监督学习

3. 在一个监督学习任务中，每个数据样本有 4 个属性和一个类别标签，每种属性分别有 3、2、2 和 2 种可能的取值，类别标签有 3 种不同的取值。请问可能有多少种不同的样本？（注意，并不是在某个数据集中最多有多少种不同的样本，而是考虑所有可能的样本）（ ）

A. 3
B. 6
C. 12
D. 24
E. 48
F. 72

该题选项 F

6. 在 Adaboosting 的迭代中，从第 t 轮到第 $t+1$ 轮，某个被错误分类样本的惩罚被增加了，可能因为该样本（ ）

A. 被第 t 轮训练的弱分类器错误分类
B. 被第 t 轮后的集成分类器（强分类器）错误分类
C. 被到第 t 轮为止训练的大多数弱分类器错误分类
D. B 和 C 都正确
E. A, B 和 C 都正确

该题选项 A

7. 假设有两批从同样的真实数据分布中采样得到去完成同一任务的数据集 A 和 B。A 包含 100K 数据，B 包含 10K 数据。按照 9:1 这一同样比例随机将 A 和 B 分别划分为训练集和测

试集。图 1 给出了数据集 A 和数据集 B 随着模型复杂度增加所对应训练误差(A,Train 以及 B,Train)和测试误差(A,Test 以及 B,Test)的曲线图。请指出哪个图正确表示了随着模型复杂度增加所对应训练误差和测试误差的变化曲线图。

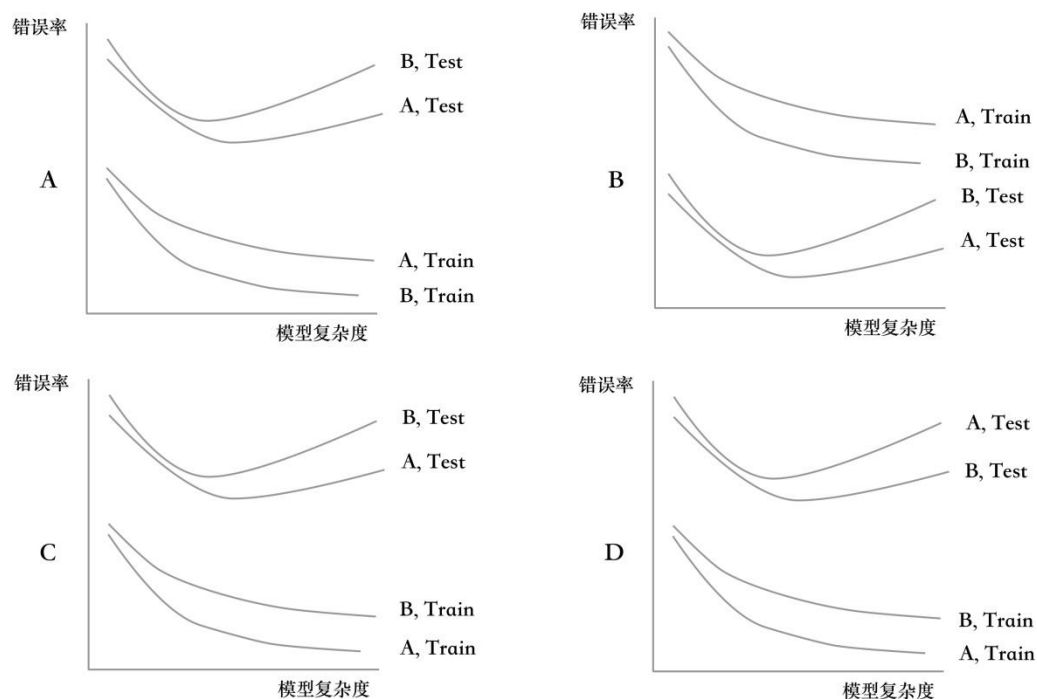


图 1 数据集 A 和数据集 B 随模型复杂度增加所对应的训练误差和测试误差变化示意图

参考答案（A 图是正确的）：

1. 对于每个数据集，随着模型复杂度增大，模型在训练集上的错误率会不断下降，而在测试集上的错误率会先下降后上升。
2. 随着模型复杂度增大，在更大的数据集 A 上模型更难拟合，因此也就不容易过拟合，但具有更好的泛化性。所以合理的猜测是，曲线（A, Train）会在（B, Train）的上方，曲线（A, Test）会在（B, Test）的下方，而曲线（A, Test）达到过拟合的转折点会比而曲线（B, Test）更靠后一些。

第五章 无监督学习

5. 在潜在语义分析中，给定 M 个单词和 N 个文档所构成的单词-文档矩阵(term-document) 矩阵，对其进行分解，将单词或文档映射到一个 R 维的隐性空间。下面描述不正确的是（ ）
 - a) 单词和文档映射到隐性空间后具有相同的维度
 - b) 通过矩阵分解可重建原始单词-文档矩阵，所得到的重建矩阵结果比原始单词-文档矩阵更好捕获了单词-单词、单词-文档、文档-文档之间的隐性关系
 - c) 这一映射过程中需要利用文档的类别信息
 - d) 隐性空间维度的大小由分解过程中所得对角矩阵中对角线上不为零的系数个数所决定
 该题选项 c

第六章 深度学习

2. 以下关于梯度下降和随机梯度下降的说明，哪些描述是正确的？

- a) 在梯度下降和随机梯度下降中，为了最小化损失函数，通常使用循环迭代的方式不断更新模型参数；
- b) 在每次迭代中，随机梯度下降需要计算训练集所有样本的误差和，用于更新模型参数；
- c) 在每次迭代中，梯度下降使用所有数据或者部分训练数据，用于更新模型参数。

该题选项：**a 和 c**

5. 图 6.4.5 展示了长短时记忆网络的详细结构。如下描述正确的是？

- a) 如果输入 x_t 为 0 向量，则 $h_t = h_{t-1}$
- b) 如果 f_t 非常小或者为 0，则误差不会被反向传播到较早的时间节点
- c) f_t 、 i_t 和 o_t 的输出是非负数；
- d) f_t 、 i_t 和 o_t 的输出可以被看作是概率分布，其输出为非负数且和为 1.

该题选项：**c**