23 春季基础算法 B Contest06

ruogu

2023 年 4 月 1 日

题目概览

哈夫曼编码

题目大意

给定 n 个字符出现的频率以及若干组对应的编码，判断每组编码是否是一个哈夫曼编码

$(1 \leq n \leq 63, m \leq 1000)$

哈夫曼编码

题目大意

给定 n 个字符出现的频率以及若干组对应的编码，判断每组编码是否是一个哈夫曼编码

$(1 \leq n \leq 63, m \leq 1000)$

- 首先需要知道哈夫曼编码的一些性质。

哈夫曼编码

题目大意

给定 n 个字符出现的频率以及若干组对应的编码，判断每组编码是否是一个哈夫曼编码

$(1 \leq n \leq 63, m \leq 1000)$

- 首先需要知道哈夫曼编码的一些性质。
- 性质一: 任何一个字符的编码不能是另一个字符编码的前缀。否则识别时将出现二义性。

哈夫曼编码

题目大意

给定 n 个字符出现的频率以及若干组对应的编码，判断每组编码是否是一个哈夫曼编码

$(1 \leq n \leq 63, m \leq 1000)$

- 首先需要知道哈夫曼编码的一些性质。
- 性质一: 任何一个字符的编码不能是另一个字符编码的前缀。否则识别时将出现二义性。
- 性质二: 对于给定频率的若干字符，任何合法哈夫曼树的 WPL 都相等。

哈夫曼编码

题目大意

给定 n 个字符出现的频率以及若干组对应的编码，判断每组编码是否是一个哈夫曼编码

$(1 \leq n \leq 63, m \leq 1000)$

- 首先需要知道哈夫曼编码的一些性质。
- 性质一: 任何一个字符的编码不能是另一个字符编码的前缀。否则识别时将出现二义性。
- 性质二: 对于给定频率的若干字符，任何合法哈夫曼树的 WPL 都相等。
- WPL:$\sum$ 每个叶子的权值 $\times$ 该叶子到根的路径长度

哈夫曼编码

- 性质一的代码实现是简单的，只需要利用 STL 中的 find 函数即可实现。

哈夫曼编码

- 性质一的代码实现是简单的，只需要利用 STL 中的 find 函数即可实现。
- 对于性质二，一个比较暴力的实现是根据输入中的哈夫曼编码建出对应的哈夫曼树并检查。

哈夫曼编码

- 性质一的代码实现是简单的，只需要利用 STL 中的 find 函数即可实现。
- 对于性质二，一个比较暴力的实现是根据输入中的哈夫曼编码建出对应的哈夫曼树并检查。
- 然而还有更精妙的实现: 考虑自底向上构建哈夫曼树的过程，我们利用小根堆每次选择两个当前权值最小的节点将其合并为一个新节点。那么只需要将过程中所有新节点的权值求和即可得到对应哈夫曼树的 WPL。

又是跑团机器人

题目大意

给定一个包含若干操作符的表达式，问表达式可计算得到的最小结果与最大结果

又是跑团机器人

题目大意

给定一个包含若干操作符的表达式，问表达式可计算得到的最小结果与最大结果

- 本题的本质是表达式计算。

又是跑团机器人

题目大意

给定一个包含若干操作符的表达式，问表达式可计算得到的最小结果与最大结果

- 本题的本质是表达式计算。
- 给我们的是中缀表达式，而中缀表达式计算起来比较困难。如果给我们的是后缀表达式，我们是不是就能够计算了呢？

又是跑团机器人

题目大意

给定一个包含若干操作符的表达式，问表达式可计算得到的最小结果与最大结果

- 本题的本质是表达式计算。
- 给我们的是中缀表达式，而中缀表达式计算起来比较困难。如果给我们的是后缀表达式，我们是不是就能够计算了呢？
- 首先需要将给定的输入分割为若干个词法单元，词法单元有两类：操作数与运算符。然后将得到的中缀表达式转化为后缀表达式。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。
- 从左到右依次扫描表达式，每次取出一个词法单元 (操作数、运算符)。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。
- 从左到右依次扫描表达式，每次取出一个词法单元 (操作数、运算符)。
- 若词法单元是操作数，则直接输出到后缀表达式中。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。
- 从左到右依次扫描表达式，每次取出一个词法单元 (操作数、运算符)。
- 若词法单元是操作数，则直接输出到后缀表达式中。
- 若词法单元是运算符，则与栈顶运算符进行比较。如果它的优先级比栈顶运算符优先级高，则直接入栈；如果它的优先级比栈顶运算符优先级低或相等，则栈顶运算符出栈并输出到后缀表达式中。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。
- 从左到右依次扫描表达式，每次取出一个词法单元 (操作数、运算符)。
- 若词法单元是操作数，则直接输出到后缀表达式中。
- 若词法单元是运算符，则与栈顶运算符进行比较。如果它的优先级比栈顶运算符优先级高，则直接入栈；如果它的优先级比栈顶运算符优先级低或相等，则栈顶运算符出栈并输出到后缀表达式中。
- 若词法单元是“(”，则直接入栈。若词法单元是“)”，则判断栈顶运算符是否为“(”。若不是，则栈顶运算符出栈，并输出到后缀表达式中，依次进行，直至栈顶运算符为“(”，抛弃“(”和“)”。

又是跑团机器人

- 首先设置一个存放运算符的栈 (运算符栈)，并置栈顶元素为“$”。“$”作为标识表达式开始的标志，另外在表达式的尾部添加一个词法单元“$”，把它作为标识表达式结束的标志。
- 从左到右依次扫描表达式，每次取出一个词法单元 (操作数、运算符)。
- 若词法单元是操作数，则直接输出到后缀表达式中。
- 若词法单元是运算符，则与栈顶运算符进行比较。如果它的优先级比栈顶运算符优先级高，则直接入栈；如果它的优先级比栈顶运算符优先级低或相等，则栈顶运算符出栈并输出到后缀表达式中。
- 若词法单元是“(”，则直接入栈。若词法单元是“)”，则判断栈顶运算符是否为“(”。若不是，则栈顶运算符出栈，并输出到后缀表达式中，依次进行，直至栈顶运算符为“(”，抛弃“(”和“)”。
- 若词法单元是“$”，则栈顶运算符依次出栈，并输出到后缀表达式中，直至栈顶运算符为"$"，抛弃“$”。

又是跑团机器人

- 得到后缀表达式后就可以计算题目要求的最小结果与最大结果了。

又是跑团机器人

- 得到后缀表达式后就可以计算题目要求的最小结果与最大结果了。
- 首先创建一个操作数栈 (栈内元素为一个 pair，代表该操作数的最小可能结果与最大可能结果)，扫描后缀表达式，如果遇到操作数则将操作数放入栈中，如果遇到运算符则将栈顶的两个操作数弹出并进行对应操作符的运算。

又是跑团机器人

- 得到后缀表达式后就可以计算题目要求的最小结果与最大结果了。
- 首先创建一个操作数栈 (栈内元素为一个 pair，代表该操作数的最小可能结果与最大可能结果)，扫描后缀表达式，如果遇到操作数则将操作数放入栈中，如果遇到运算符则将栈顶的两个操作数弹出并进行对应操作符的运算。
- 对于每个操作符 (以 * 为例)，我们需要讨论其运算之后的最小结果与最大结果，可以通过枚举
  $Min * Min, Min * Max, Max * Min, Max * Max$ 的四种结果并取 Min 与 Max 即可，需要注意的是对 d 运算需要特别讨论。

又是跑团机器人

- 得到后缀表达式后就可以计算题目要求的最小结果与最大结果了。
- 首先创建一个操作数栈 (栈内元素为一个 pair，代表该操作数的最小可能结果与最大可能结果)，扫描后缀表达式，如果遇到操作数则将操作数放入栈中，如果遇到运算符则将栈顶的两个操作数弹出并进行对应操作符的运算。
- 对于每个操作符 (以 * 为例)，我们需要讨论其运算之后的最小结果与最大结果，可以通过枚举 $Min * Min, Min * Max, Max * Min, Max * Max$ 的四种结果并取 Min 与 Max 即可，需要注意的是对 d 运算需要特别讨论。
- 最终栈顶元素即为所求的最小结果与最大结果。

前世档案

题目大意

给定 N 个问题与 M 个玩家，分别告诉你每个玩家对于这 N 个问题的回答，问这个玩家对应的结论编号

$(1 \leq N \leq 30, M \leq 100)$

前世档案

题目大意

给定 N 个问题与 M 个玩家，分别告诉你每个玩家对于这 N 个问题的回答，问这个玩家对应的结论编号
$(1 \leq N \leq 30, M \leq 100)$

- 由于回答只有两种，故可以构造一颗决策树，其是一颗满二叉树且每个叶子节点均为一个最后的结论。

前世档案

题目大意

给定 N 个问题与 M 个玩家，分别告诉你每个玩家对于这 N 个问题的回答，问这个玩家对应的结论编号
$(1 \leq N \leq 30, M \leq 100)$

- 由于回答只有两种，故可以构造一颗决策树，其是一颗满二叉树且每个叶子节点均为一个最后的结论。
- 那么我们可以按照回答模拟出最终到达叶节点的编号，并预先计算出最靠左的叶子节点的编号，两者做差即可得到结论的编号。

前世档案

题目大意

给定 N 个问题与 M 个玩家，分别告诉你每个玩家对于这 N 个问题的回答，问这个玩家对应的结论编号

$(1 \leq N \leq 30, M \leq 100)$

- 由于回答只有两种，故可以构造一颗决策树，其是一颗满二叉树且每个叶子节点均为一个最后的结论。
- 那么我们可以按照回答模拟出最终到达叶节点的编号，并预先计算出最靠左的叶子节点的编号，两者做差即可得到结论的编号。
- 时间复杂度 $O(NM)$。

奶牛的区间游戏

题目大意

n 个区间，第 i 个区间以 a_i 为起点，b_i 为终点，且满足 $0 \le a_i \le b_i \le m$。Bessie 与 Elsie 分别选择两个区间 (这两个区间可能相同)。对于给定的 k，假设 Bessie 与 Elsie 选中的区间分别为第 i 个与第 j 个区间，若满足 $a_i + a_j \le k \le b_i + b_j$，则她们获胜。

对 0 到 $2 \times m$ 中的每一个 k，计算使得她们获胜的有序对 (i,j) 的数量。

$(1 \le n \le 2 \times 10^5, 1 \le m \le 5000)$

奶牛的区间游戏

题目大意

n 个区间，第 i 个区间以 a_i 为起点，b_i 为终点，且满足 $0 \le a_i \le b_i \le m$。Bessie 与 Elsie 分别选择两个区间 (这两个区间可能相同)。对于给定的 k，假设 Bessie 与 Elsie 选中的区间分别为第 i 个与第 j 个区间，若满足 $a_i + a_j \le k \le b_i + b_j$，则她们获胜。

对 0 到 $2 \times m$ 中的每一个 k，计算使得她们获胜的有序对 (i,j) 的数量。

$(1 \le n \le 2 \times 10^5, 1 \le m \le 5000)$

- 如果我们枚举 k、i、j 来计算答案，时间复杂度为 $O(n^2 \times m)$，只能得到 20 分。

奶牛的区间游戏

题目大意

n 个区间，第 i 个区间以 a_i 为起点，b_i 为终点，且满足 $0 \le a_i \le b_i \le m$。Bessie 与 Elsie 分别选择两个区间 (这两个区间可能相同)。对于给定的 k，假设 Bessie 与 Elsie 选中的区间分别为第 i 个与第 j 个区间，若满足 $a_i + a_j \le k \le b_i + b_j$，则她们获胜。

对 0 到 $2 \times m$ 中的每一个 k，计算使得她们获胜的有序对 (i,j) 的数量。

$(1 \le n \le 2 \times 10^5, 1 \le m \le 5000)$

- 如果我们枚举 k、i、j 来计算答案，时间复杂度为 $O(n^2 \times m)$，只能得到 20 分。
- 看来从 k 入手统计是不行的，我们换一种思路，假设固定了 i 与 j，这个有序对能对哪些 k 产生贡献呢？

奶牛的区间游戏

题目大意

n 个区间，第 i 个区间以 a_i 为起点，b_i 为终点，且满足 $0 \le a_i \le b_i \le m$。Bessie 与 Elsie 分别选择两个区间 (这两个区间可能相同)。对于给定的 k，假设 Bessie 与 Elsie 选中的区间分别为第 i 个与第 j 个区间，若满足 $a_i + a_j \le k \le b_i + b_j$，则她们获胜。

对 0 到 $2 \times m$ 中的每一个 k，计算使得她们获胜的有序对 (i,j) 的数量。

$(1 \le n \le 2 \times 10^5, 1 \le m \le 5000)$

- 如果我们枚举 k、i、j 来计算答案，时间复杂度为 $O(n^2 \times m)$，只能得到 20 分。
- 看来从 k 入手统计是不行的，我们换一种思路，假设固定了 i 与 j，这个有序对能对哪些 k 产生贡献呢？
- 由于 i 与 j 固定，那么不难发现对于所有满足 $a_i + a_j \le k \le b_i + b_j$ 的 k，这个有序对都能对 k 产生贡献。这对我们有什么帮助？

奶牛的区间游戏

题目大意

n 个区间，第 i 个区间以 a_i 为起点，b_i 为终点，且满足 $0 \le a_i \le b_i \le m$。Bessie 与 Elsie 分别选择两个区间 (这两个区间可能相同)。对于给定的 k，假设 Bessie 与 Elsie 选中的区间分别为第 i 个与第 j 个区间，若满足 $a_i + a_j \le k \le b_i + b_j$，则她们获胜。

对 0 到 $2 \times m$ 中的每一个 k，计算使得她们获胜的有序对 (i,j) 的数量。

$(1 \le n \le 2 \times 10^5, 1 \le m \le 5000)$

- 如果我们枚举 k、i、j 来计算答案，时间复杂度为 $O(n^2 \times m)$，只能得到 20 分。
- 看来从 k 入手统计是不行的，我们换一种思路，假设固定了 i 与 j，这个有序对能对哪些 k 产生贡献呢？
- 由于 i 与 j 固定，那么不难发现对于所有满足 $a_i + a_j \le k \le b_i + b_j$ 的 k，这个有序对都能对 k 产生贡献。这对我们有什么帮助？
- 对于固定的 i 与 j，其产生贡献的区间就是 $[a_i + a_j, b_i + b_j]$！

奶牛的区间游戏

- 但如果我们遍历 $[a_i + a_j, b_i + b_j]$ 来维护答案，那么复杂度还是没变呀！怎样优化对答案的统计呢？

奶牛的区间游戏

- 但如果我们遍历 $[a_i + a_j, b_i + b_j]$ 来维护答案，那么复杂度还是没变呀！怎样优化对答案的统计呢？
- 这实际上是对给定的区间的 k 答案整体 $+1$，因此我们可以使用差分。

奶牛的区间游戏

- 但如果我们遍历 $[a_i + a_j, b_i + b_j]$ 来维护答案，那么复杂度还是没变呀！怎样优化对答案的统计呢？
- 这实际上是对给定的区间的 k 答案整体 $+1$，因此我们可以使用差分。
- 维护一个差分数组 d，对区间 $[a_i + a_j, b_i + b_j]$ 整体 $+1$ 即为 d$[a_i + a_j]$+=1,d$[b_i + b_j + 1]$-=1。

奶牛的区间游戏

- 但如果我们遍历 $[a_i + a_j, b_i + b_j]$ 来维护答案，那么复杂度还是没变呀！怎样优化对答案的统计呢？
- 这实际上是对给定的区间的 k 答案整体 $+1$，因此我们可以使用差分。
- 维护一个差分数组 d，对区间 $[a_i + a_j, b_i + b_j]$ 整体 $+1$ 即为 d$[a_i + a_j]$+=1,d$[b_i + b_j + 1]$-=1。
- 最后通过对差分数组求前缀和即可知道每个 k 的答案，时间复杂度 $O(n^2)$，可以得到 50 分。

奶牛的区间游戏

- 注意到数据范围中 $1 \leq m \leq 5000$ 这个条件，我们能不能利用这个条件进一步降低算法的时间复杂度？

奶牛的区间游戏

- 注意到数据范围中 $1 \leq m \leq 5000$ 这个条件，我们能不能利用这个条件进一步降低算法的时间复杂度？
- 由于我们在差分的时候只需要令 d[$a_i + a_j$]+=1,d[$b_i + b_j + 1$]-=1。我们换一种思路考虑，对于给定的某个数 x，有多少组 $a_i + a_j$ 会令 d[x]+=1，有多少组 $b_i + b_j$ 会令 d[x]-=1 呢？

奶牛的区间游戏

- 注意到数据范围中 $1 \leq m \leq 5000$ 这个条件，我们能不能利用这个条件进一步降低算法的时间复杂度？
- 由于我们在差分的时候只需要令 d[$a_i + a_j$]+=1,d[$b_i + b_j + 1$]-=1。我们换一种思路考虑，对于给定的某个数 x，有多少组 $a_i + a_j$ 会令 d[x]+=1，有多少组 $b_i + b_j$ 会令 d[x]-=1 呢？
- 只有当某个有序对满足 $a_i + a_j = x$ 时会令 d[x]+=1，满足 $b_i + b_j = x - 1$ 时会令 d[x]-=1！

奶牛的区间游戏

- 注意到数据范围中 $1 \leq m \leq 5000$ 这个条件，我们能不能利用这个条件进一步降低算法的时间复杂度？
- 由于我们在差分的时候只需要令 d[$a_i + a_j$]+=1,d[$b_i + b_j + 1$]-=1。我们换一种思路考虑，对于给定的某个数 x，有多少组 $a_i + a_j$ 会令 d[x]+=1，有多少组 $b_i + b_j$ 会令 d[x]-=1 呢？
- 只有当某个有序对满足 $a_i + a_j = x$ 时会令 d[x]+=1，满足 $b_i + b_j = x - 1$ 时会令 d[x]-=1！
- 所以我们其实只需要统计有多少对 (i,j) 满足 $a_i + a_j = x$ 以及 $b_i + b_j = x$。

奶牛的区间游戏

- 结合 $1 \leq m \leq 5000$ 这个条件，我们可以用桶计数，通过枚举值域来进行统计。

奶牛的区间游戏

- 结合 $1 \leq m \leq 5000$ 这个条件，我们可以用桶计数，通过枚举值域来进行统计。
- 具体的，建立 cnt1 与 cnt2 两个桶，cnt1[i] 表示满足 $a_j = i$ 的下标 j 有 cnt1[i] 个，cnt2[i] 表示满足 $b_j = i$ 的下标 j 有 cnt2[i] 个。

奶牛的区间游戏

- 结合 $1 \leq m \leq 5000$ 这个条件，我们可以用桶计数，通过枚举值域来进行统计。
- 具体的，建立 cnt1 与 cnt2 两个桶，cnt1[i] 表示满足 $a_j = i$ 的下标 j 有 cnt1[i] 个，cnt2[i] 表示满足 $b_j = i$ 的下标 j 有 cnt2[i] 个。
- 我们两重循环枚举 a_i 与 a_j 的值域 p 与 q，则满足 $a_i + a_j = p + q$ 的有序对 (i,j) 个数为 $cnt1[p] \times cnt1[q]$，那么对于差分数组我们只需要令 $d[p+q] += cnt1[p] \times cnt1[q]$ 即可。

奶牛的区间游戏

- 结合 $1 \leq m \leq 5000$ 这个条件，我们可以用桶计数，通过枚举值域来进行统计。
- 具体的，建立 cnt1 与 cnt2 两个桶，cnt1[i] 表示满足 $a_j = i$ 的下标 j 有 cnt1[i] 个，cnt2[i] 表示满足 $b_j = i$ 的下标 j 有 cnt2[i] 个。
- 我们两重循环枚举 a_i 与 a_j 的值域 p 与 q，则满足 $a_i + a_j = p + q$ 的有序对 (i,j) 个数为 $cnt1[p] \times cnt1[q]$，那么对于差分数组我们只需要令 $d[p+q] += cnt1[p] \times cnt1[q]$ 即可。
- 同样两重循环枚举 b_i 与 b_j 的值域 p 与 q，则满足 $b_i + b_j = p + q$ 的有序对 (i,j) 个数为 $cnt2[p] \times cnt2[q]$，那么对于差分数组我们只需要令 $d[p+q+1] -= cnt2[p] \times cnt2[q]$ 即可。

奶牛的区间游戏

- 结合 $1 \leq m \leq 5000$ 这个条件，我们可以用桶计数，通过枚举值域来进行统计。
- 具体的，建立 cnt1 与 cnt2 两个桶，cnt1[i] 表示满足 $a_j = i$ 的下标 j 有 cnt1[i] 个，cnt2[i] 表示满足 $b_j = i$ 的下标 j 有 cnt2[i] 个。
- 我们两重循环枚举 a_i 与 a_j 的值域 p 与 q，则满足 $a_i + a_j = p + q$ 的有序对 (i,j) 个数为 $cnt1[p] \times cnt1[q]$，那么对于差分数组我们只需要令 $d[p+q] += cnt1[p] \times cnt1[q]$ 即可。
- 同样两重循环枚举 b_i 与 b_j 的值域 p 与 q，则满足 $b_i + b_j = p + q$ 的有序对 (i,j) 个数为 $cnt2[p] \times cnt2[q]$，那么对于差分数组我们只需要令 $d[p+q+1] -= cnt2[p] \times cnt2[q]$ 即可。
- 顺利求得 d 数组后求前缀和即可得到答案，时间复杂度 $O(m^2)$。

The End

谢谢大家