

NOIP 模拟赛试题讨论

wzj52501

2018 年 10 月 27 日

Results

得分统计

- 随机输出"Yes" 或"No"。

- 随机输出"Yes" 或"No"。
- 哦，你得了满分（逃

- 枚举每个友方随从攻击了哪一个敌方随从，然后检验即可。

- 枚举每个友方随从攻击了哪一个敌方随从，然后检验即可。
- 实现时需要注意很多细节，如随从可以不攻击、攻击力为 0 的随从无法攻击等。

- 枚举每个友方随从攻击了哪一个敌方随从，然后检验即可。
- 实现时需要注意很多细节，如随从可以不攻击、攻击力为 0 的随从无法攻击等。
- 时间复杂度为 $O(nm \times m^n)$ 。

得分统计

- BFS 判断起点到唯一宝藏点距离是否相等即可。

- BFS 判断起点到唯一宝藏点距离是否相等即可。
- 时间复杂度为 $O(n \times m)$ 。

- 每次我们肯定是从某个宝藏点或起点出发，沿最短路径到达另一个宝藏点。

- 每次我们肯定是从某个宝藏点或起点出发，沿最短路径到达另一个宝藏点。
- 我们可以全排列枚举访问顺序，依次判断最多访问到哪一处宝藏。

- 每次我们肯定是从某个宝藏点或起点出发，沿最短路径到达另一个宝藏点。
- 我们可以全排列枚举访问顺序，依次判断最多访问到哪一处宝藏。
- 预处理每个宝藏间的最短距离和每个宝藏到起点的最短距离，时间复杂度为 $O(k \times m \times n + (k + 1)!)$ 。

- 我们可以在上面算法的基础上进一步改进，为了方便把起点作为 0 号宝藏点。设 $f(i, S)$ 表示当前在 i 号宝藏点，且已经到达过 S 集合内的宝藏点的最短时间。

- 我们可以在上面算法的基础上进一步改进，为了方便把起点作为 0 号宝藏点。设 $f(i, S)$ 表示当前在 i 号宝藏点，且已经到达过 S 集合内的宝藏点的最短时间。
- 转移时枚举下一次去哪个宝藏点， $f(i, S) + dist(i, j) \rightarrow f(j, S \cup \{j\})$ 。

- 我们可以在上面算法的基础上进一步改进，为了方便把起点作为 0 号宝藏点。设 $f(i, S)$ 表示当前在 i 号宝藏点，且已经到达过 S 集合内的宝藏点的最短时间。
- 转移时枚举下一次去哪个宝藏点， $f(i, S) + dist(i, j) \rightarrow f(j, S \cup \{j\})$ 。
- 初始只有 $f(0, \{0\})$ 为 0，其余均设为无穷大，最后在所有 $f(j, S) \leq T$ 的 S 中选出 $|S|$ 最大的集合即为答案。

- 我们可以在上面算法的基础上进一步改进，为了方便把起点作为 0 号宝藏点。设 $f(i, S)$ 表示当前在 i 号宝藏点，且已经到达过 S 集合内的宝藏点的最短时间。
- 转移时枚举下一次去哪个宝藏点， $f(i, S) + dist(i, j) \rightarrow f(j, S \cup \{j\})$ 。
- 初始只有 $f(0, \{0\})$ 为 0，其余均设为无穷大，最后在所有 $f(j, S) \leq T$ 的 S 中选出 $|S|$ 最大的集合即为答案。
- 时间复杂度为 $O(k \times m \times n + k^2 \times 2^k)$ 。

得分统计

不说了...

- 对于这一部分数据, $n \leq 2501$ 。

- 对于这一部分数据， $n \leq 2501$ 。
- 设 f_i 表示前 i 个音符已经划分完毕，已经划分过的最短乐章最长是多少，易得 $f_i = \max \{ \min(f_j, i - j) \}$ ，其中 $[j + 1, i]$ 区间没有相同的音符。

- 对于这一部分数据， $n \leq 2501$ 。
- 设 f_i 表示前 i 个音符已经划分完毕，已经划分过的最短乐章最长是多少，易得 $f_i = \max \{ \min(f_j, i - j) \}$ ，其中 $[j + 1, i]$ 区间没有相同的音符。
- 计算方案的话可以设 g_i 表示前 i 个音符已经划分完毕，已经划分过的乐章长度均 $\geq f_n$ ，可得 $g_i = \sum_{j=0}^{i-f_n} g_j$ ，其中 $[j + 1, i]$ 区间没有相同的音符。

- 对于这一部分数据， $n \leq 2501$ 。
- 设 f_i 表示前 i 个音符已经划分完毕，已经划分过的最短乐章最长是多少，易得 $f_i = \max \{ \min(f_j, i - j) \}$ ，其中 $[j + 1, i]$ 区间没有相同的音符。
- 计算方案的话可以设 g_i 表示前 i 个音符已经划分完毕，已经划分过的乐章长度均 $\geq f_n$ ，可得 $g_i = \sum_{j=0}^{i-f_n} g_j$ ，其中 $[j + 1, i]$ 区间没有相同的音符。
- 答案即为 f_n 和 g_n ，时间复杂度为 $O(n^2)$ 。

- 我们先对于每个右端点 r , 求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。

- 我们先对于每个右端点 r , 求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。

- 我们先对于每个右端点 r , 求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。
- 所以我们可以利用前缀和在 $O(n)$ 的时间内计算出 g 数组 , 那么瓶颈就在于求 f_n 。

- 我们先对于每个右端点 r ，求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。
- 所以我们可以利用前缀和在 $O(n)$ 的时间内计算出 g 数组，那么瓶颈就在于求 f_n 。
- 考虑二分答案，如何判断是否存在一个方案使得每段乐章长度均 $\geq x$ 呢？

- 我们先对于每个右端点 r ，求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。
- 所以我们可以利用前缀和在 $O(n)$ 的时间内计算出 g 数组，那么瓶颈就在于求 f_n 。
- 考虑二分答案，如何判断是否存在一个方案使得每段乐章长度均 $\geq x$ 呢？
- 我们只需要再计算一遍 g 数组，然后判断 g_n 是否大于 0 就可以了！

- 我们先对于每个右端点 r , 求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。
- 所以我们可以利用前缀和在 $O(n)$ 的时间内计算出 g 数组, 那么瓶颈就在于求 f_n 。
- 考虑二分答案, 如何判断是否存在一个方案使得每段乐章长度均 $\geq x$ 呢?
- 我们只需要再计算一遍 g 数组, 然后判断 g_n 是否大于 0 就可以了!
- 其实第二问就是用来提示第一问的 ($>$ $<$)

- 我们先对于每个右端点 r , 求出极小左端点 $left_r$ 满足 $[left_r, r]$ 区间没有相同的音符。
- 可以递推求出 $left_r = \max(left_{r-1}, lastpos(A_r))$ 。
- 所以我们可以利用前缀和在 $O(n)$ 的时间内计算出 g 数组, 那么瓶颈就在于求 f_n 。
- 考虑二分答案, 如何判断是否存在一个方案使得每段乐章长度均 $\geq x$ 呢?
- 我们只需要再计算一遍 g 数组, 然后判断 g_n 是否大于 0 就可以了!
- 其实第二问就是用来提示第一问的 ($>$ $<$)
- 时间复杂度为 $O(n \log n)$ 。

Q & A