Day 7 solution

faebdc

2016 年 5 月 29 日

数值修改

100分：

数值修改

算法1

数值修改

算法1

f_i表示i最少经过多少次变成0。进行dp。

数值修改

算法1

f_i表示i最少经过多少次变成0。进行dp。

时间复杂度：$O(n)$

数值修改

算法1

f_i表示i最少经过多少次变成0。进行dp。

时间复杂度：$O(n)$

期望得分：30分

数值修改

算法2

数值修改

算法2

可以发现，每次减去最大的数一定是最优的。

数值修改

算法2

可以发现，每次减去最大的数一定是最优的。

模拟这个过程。

数值修改

算法2

可以发现，每次减去最大的数一定是最优的。

模拟这个过程。

时间复杂度：$O(n)$

数值修改

算法2

可以发现，每次减去最大的数一定是最优的。

模拟这个过程。

时间复杂度：$O(n)$

期望得分：30分

数值修改

算法3

数值修改

算法3

把数的前6位与后6位分开考虑。前6位的变化次数是$O(\sqrt{n})$的。我们需要快速计算前6位在几次之后发生变化，并且需要得到变化之后的数字。

数值修改

算法3

把数的前6位与后6位分开考虑。前6位的变化次数是$O(\sqrt{n})$的。我们需要快速计算前6位在几次之后发生变化，并且需要得到变化之后的数字。

前6位中的最大值相同时，后6位的这些变化是相同的。所以对于每一个前6位中的最大值，都预处理一下就可以了。

数值修改

算法3

把数的前6位与后6位分开考虑。前6位的变化次数是$O(\sqrt{n})$的。我们需要快速计算前6位在几次之后发生变化，并且需要得到变化之后的数字。

前6位中的最大值相同时，后6位的这些变化是相同的。所以对于每一个前6位中的最大值，都预处理一下就可以了。

时间复杂度：$O(\sqrt{n})$

数值修改

算法3

把数的前6位与后6位分开考虑。前6位的变化次数是$O(\sqrt{n})$的。我们需要快速计算前6位在几次之后发生变化，并且需要得到变化之后的数字。

前6位中的最大值相同时，后6位的这些变化是相同的。所以对于每一个前6位中的最大值，都预处理一下就可以了。

时间复杂度：$O(\sqrt{n})$

期望得分：60分

数值修改

算法4

数值修改

算法4

定义$f[i][j][k]$表示i位数，前$i-1$为是9，个位是j，每次操作若各个数位上的数都小于k，则减去k，否则减去各个数位上的数中的最大值，将数减成负数需要多少次，同时记录那个负数是几。

数值修改

算法4

定义$f[i][j][k]$表示i位数，前$i-1$为是9，个位是j，每次操作若各个数位上的数都小于k，则减去k，否则减去各个数位上的数中的最大值，将数减成负数需要多少次，同时记录那个负数是几。

初始化f数组后通过f数组的值计算答案即可。

数值修改

算法4

定义$f[i][j][k]$表示i位数，前$i-1$为是9，个位是j，每次操作若各个数位上的数都小于k，则减去k，否则减去各个数位上的数中的最大值，将数减成负数需要多少次，同时记录那个负数是几。

初始化f数组后通过f数组的值计算答案即可。

时间复杂度：$O(log\ \ n)$

数值修改

算法4

定义$f[i][j][k]$表示i位数，前$i-1$为是9，个位是j，每次操作若各个数位上的数都小于k，则减去k，否则减去各个数位上的数中的最大值，将数减成负数需要多少次，同时记录那个负数是几。

初始化f数组后通过f数组的值计算答案即可。

时间复杂度：$O(log\ n)$

期望得分：100分

覆盖

100分：

覆盖

算法1

覆盖

算法1

路径两两枚举，暴力判断。

覆盖

算法1

路径两两枚举，暴力判断。

时间复杂度：$O(n \cdot m^2)$

覆盖

算法1

路径两两枚举，暴力判断。

时间复杂度：$O(n \cdot m^2)$

期望得分：30分

覆盖

算法2

覆盖

算法2

若a在c的子树内，b在d的子树内，那么(a, b)就覆盖了(c, d)。

覆盖

算法2

若a在c的子树内，b在d的子树内，那么(a, b)就覆盖了(c, d)。

可以用dfs序判断是否在子树内。

覆盖

算法2

若a在c的子树内，b在d的子树内，那么(a,b)就覆盖了(c,d)。

可以用dfs序判断是否在子树内。

路径两两枚举，借助dfs序判断。

覆盖

算法2

若a在c的子树内，b在d的子树内，那么(a, b)就覆盖了(c, d)。

可以用dfs序判断是否在子树内。

路径两两枚举，借助dfs序判断。

时间复杂度：$O(m^2)$

覆盖

算法2

若a在c的子树内，b在d的子树内，那么(a, b)就覆盖了(c, d)。

可以用dfs序判断是否在子树内。

路径两两枚举，借助dfs序判断。

时间复杂度：$O(m^2)$

期望得分：60分

覆盖

算法3

覆盖

算法3

对于一条路径(a, b)，求有多少条路径覆盖了它，也就是求有多少路径的一端在a的子树中，一端在b的子树中。

覆盖

算法3

对于一条路径(a, b)，求有多少条路径覆盖了它，也就是求有多少路径的一端在a的子树中，一端在b的子树中。

这就相当于是求矩形内点的个数。

覆盖

算法3

对于一条路径(a, b)，求有多少条路径覆盖了它，也就是求有多少路径的一端在a的子树中，一端在b的子树中。

这就相当于是求矩形内点的个数。

可以用扫描线的方法求出。

覆盖

算法3

对于一条路径(a, b)，求有多少条路径覆盖了它，也就是求有多少路径的一端在a的子树中，一端在b的子树中。

这就相当于是求矩形内点的个数。

可以用扫描线的方法求出。

时间复杂度：$O(n\ log\ n)$

覆盖

算法3

对于一条路径(a, b)，求有多少条路径覆盖了它，也就是求有多少路径的一端在a的子树中，一端在b的子树中。

这就相当于是求矩形内点的个数。

可以用扫描线的方法求出。

时间复杂度：$O(n\ log\ n)$

期望得分：100分

机器人

100分：

机器人

算法1

机器人

算法1

模拟

机器人

算法1

模拟

时间复杂度：$O(nm)$

机器人

算法1

模拟

时间复杂度：$O(nm)$

期望得分：20分

机器人

算法2

机器人

算法2

速度只会是-1、0、1时，对三个速度各自维护最大、最小值。

机器人

算法2

速度只会是-1、0、1时，对三个速度各自维护最大、最小值。

时间复杂度：$O(m\ log\ n)$

机器人

算法2

速度只会是-1、0、1时，对三个速度各自维护最大、最小值。

时间复杂度：$O(m\ log\ n)$

期望得分：10分

机器人

算法3

机器人

算法3

机器人碰面次数较少时，可以用平衡树维护机器人先后顺序。

机器人

算法3

机器人碰面次数较少时，可以用平衡树维护机器人先后顺序。

用堆记录相邻机器人下一次碰面的时间，每次取最早的时间调整平衡树中的顺序。

机器人

算法3

机器人碰面次数较少时，可以用平衡树维护机器人先后顺序。

用堆记录相邻机器人下一次碰面的时间，每次取最早的时间调整平衡树中的顺序。

时间复杂度：$O(4 \times 10^5 \ log \ n)$

机器人

算法3

机器人碰面次数较少时，可以用平衡树维护机器人先后顺序。

用堆记录相邻机器人下一次碰面的时间，每次取最早的时间调整平衡树中的顺序。

时间复杂度：$O(4 \times 10^5 \ log \ n)$

期望得分：20分

机器人

算法4

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

合并果子

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

合并果子　哈夫曼树

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

合并果子　　哈夫曼树

每次选取段数最少的两个函数合并。

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

合并果子　哈夫曼树

每次选取段数最少的两个函数合并。

时间复杂度：$O((n+m)\ log\ n)$

机器人

算法4

每个机器人的位置是关于时间的分段函数。我们关心的是这些分段函数在每个时刻的最大值和最小值。下面考虑如何求最大值，最小值的求法类似。

对于一个a段的函数和一个b段的函数，可以以$O(a+b)$的时间复杂度将它们合并成一个$O(a+b)$段的函数。

最少次数是多少？

合并果子　哈夫曼树

每次选取段数最少的两个函数合并。

时间复杂度：$O((n+m)\ log\ n)$

期望得分：100分

Q&A