1 有趣的数

1.1 60 分做法

枚举 $[L,R]$ 中的每个数，逐个判定，时间复杂度 $\mathcal{O}(R\log_{10}R)$。

1.2 100 分做法

尝试生成所有的数，枚举这个数的位数、独特的数字和其余的数字以及独特数字的位置，同时判断是否 $\in[X,Y]$。时间复杂度 $\mathcal{O}(17^2\cdot 9^2)$。

2 欢乐 ABC

2.1 30 分做法

暴力枚举两个端点，并扫描并统计之间的字符，然后判断计入答案。$O(n^3)$。

2.2 70 分做法

分别使用三个前缀和数组 $A[i],B[i],C[i]$ 表示 $1...i$ 之间的 A,B,C 字符个数。这样直接枚举两个端点 i,j，中间的字符数就可以通过 $A[j]-A[i-1]$（B,C 一样）可以获得。$O(n^2)$。

2.3 100 分做法

我们可以发现当 $S[i..j]$ 是一个合法的非空子串的时候，当且仅当 $A[i-1]-B[i-1]==A[j]-B[j]$ 且 $B[i-1]-C[i-1]==B[j]-C[j]$，那么对于所有的 $i(0\le i\le n)$，把 $(A[i]-B[i],B[i]-C[i])$ 看成一个二元组，然后统计有多少对相同的组合即可。用 map 边加边统计，或者也可以排好序以后看连续相等的，然后算进答案里面。

3 数位问题

3.1 10 分做法

容易知道最终的数里面没有相邻的两个 0，这样的话答案不超过 $2\sum d_i$，因此暴力枚举每一位是几并判断。

时间复杂度 $\mathcal{O}(T\cdot 2\sum d_i 10^{2\sum d_i})$。

3.2 30 分做法

稍微优化一下上面的暴力，考虑选取 $2\sum d_i$ 个位置中的 $\sum d_i$ 个放置（注意全排列）。

时间复杂度 $\mathcal{O}\left(T\frac{(2\sum d_i)!}{(\sum d_i)!}\right)$。

3.3 60 分做法

一个数被 11 整除的充要条件是：这个数奇数位上数字和 − 偶数位上数字和被 11 整除。

因此考虑枚举每个数字放在奇数位还是偶数位，其余用 0 补齐。

时间复杂度 $\mathcal{O}(T \cdot 2^{\sum d_i})$。

3.4 100 分做法

采用动态规划算法。

容易知道数字顺序和答案无关，那么我们按照 $1 \sim 9$ 的顺序依次加入。

$f[i][j][k]$ 表示前 $1 \sim i$ 都全部加入后，奇数位和 − 偶数位和除以 11 的余数是 j，有 k 个数字被放在了奇数位上，这种情况可不可行。

转移非常方便，不再赘述。

时间复杂度 $\mathcal{O}(T \cdot 9 \cdot 11 \cdot (\sum d_i)^2)$。

注意无解的情况，不然过不了前两个测试点。

4 打游戏

4.1 测试点 1、2

此时不能放技能，只能一个个打。容易证明将怪按照生命值升序排序然后逐个歼灭答案最优。

时间复杂度 $\mathcal{O}(n \log n)$。

4.2 测试点 3

此时能放一次技能，容易证明先放技能答案最优。枚举放哪个技能，后面的就按照生命值排序逐个歼灭。

时间复杂度 $\mathcal{O}(n \log n)$。

4.3 测试点 4

延续上面的思路，这 m 个技能一定在前 m 个回合释放，枚举每回合放哪个技能（重击的话依然对生命值最小的放），每次完后都要暴力修改数组。

时间复杂度（上限）$\mathcal{O}(n \log n + n \cdot 2^m)$。

4.4 测试点 5

可以发现每次枚举完技能后没必要暴力修改，对于群攻，记录一下次数即可，对于重击某个怪，暴力修改。

时间复杂度 $\mathcal{O}(n\log n+2^m)$。

4.5 测试点 6、7、8

采用动态规划算法。设 $f[i][c][z][q]$ 表示现在有 i 点魔法值，正在打第 c 个怪（前 $c-1$ 个都打完了），对这个怪用了 z 次重击，一共用过 q 次群攻，从现在到结束最少要多少生命。

可以得到 $f[i][n+1][z][q]=0$，答案是 $f[m][1][0][0]$。

当前怪的生命值是 h_i-2z-q，如果选择重击，那么从 $f[i-1][c][z+1][q]$ 转移，如果选择群攻，从 $f[i-1][c][z][q+1]$ 转移。

时间复杂度 $\mathcal{O}(n\log n+nm^2)$，空间可以用滚动数组优化到 $\mathcal{O}(nm)$。

4.6 满分做法

玩过游戏的同学肯定都会想到：先把生命从小到大排好序，然后依次去打，如果当前怪的生命值为 1 或场上不止两个怪，必然放群攻，否则放重击。如果魔法值用完了就一个个普通攻击。

显然是正确的，当你场上最小的怪生命值为 1 时显然群攻。

其次当你不止两个怪的时候，因为群攻一下可以削减 3 点生命值，显然赚（因为一次重击会提早一轮把最小的怪砍死，你有概率会少吃一点伤害，但是你会损失一整个轮次，那样你可能会受到 1 及 1 以上的伤害，所以不赚）。

显然只有两个怪的时候且生命值 ≥ 2 的时候从小到大重击。

时间复杂度 $\mathcal{O}(n\log n)$。