白银之春

题目翻译：给你 N 个整数，要求从中选不超过 K 段连续的数字，要求其相加之和最大。

10 分做法：对每个数字枚举选或不选，复杂度 $O(2^n)$。

100 分做法：DP，假设当前处理到了第 i 个数字。先仅以f_i表示从开头到第 i 位的最大值。f_i必定从$f_j(1 \le j < i)$转移而来，但是这样就无法处理选择数字的段数。所以我们可以再加上一维，这一维与段数相关。

这样，最简单的方法就是用$f_{i,x}$表示处理到了第 i 个数字时，选择不超过 x 段数字的最大值。我们仍然考虑从选和不选两个角度对其进行处理。

不选第 i 个数字，那么假设当前已经选择了不超过 x 段数字，到第 i 位时选择的数字依然为不超过 x 段，即$f_{i,x} = \max\left(f_{j,x}\right)(1 \le j < i)$。

选第 i 个数字，则其必被并在某一段连续数字中。显然，第 i 个数字所处的这一段必定为当前选择的最后一段连续数字。由于前 $i-1$ 个数字的选择情况未知，$f_{i,x}$不能简单地从$f_{j,x}(1 \le j < i)$转移而来，应采取枚举起点的方法确定$f_{i,x}$。

不妨令起点为第 k 个数字，枚举 k 的取值，那么这就相当于新加了一段数字。提前处理数字的前缀和为s_i，则$f_{i,x} = \max\left(f_{k-1,x-1} + s_i - s_{k-1}\right)(1 \le k \le i)$。

所以，动态转移方程为：

$$f_{i,x} = \begin{cases} \max\left(f_{j,x}\right)(1 \le j < i) \\ \max\left(f_{k-1,x-1} + s_i - s_{k-1}\right)(1 \le k \le i) \end{cases}$$

复杂度为 $O(n^3)$。

从转移的形式可看出$f_{i,x} \ge f_{i-1,x} \ge \max\left(f_{j,x}\right)(1 \le j < i)$，所以直接将$f_{i,x}$赋值为$f_{i-1,x}$省去第一种情况。设$g_{i,x} = \max\left(f_{j,x} - s_j\right)(1 \le j \le i)$，那么第二种情况可以写作$f_{i,x} = g_{i-1,x-1} + s_i$。

如此，可以优化到 $O(n^2)$，也许应该要求灵梦加钱？

迷途之家

题目翻译：给你一个 N 个点，M 条边的无向无权图，求从 K 到 D 的最短路径数量。

10 分做法：我也不清楚，不过各位天资过人，应该是能想出完美的暴力卡过这种数据的。

100 分做法：假设这不是一个点和线组成的图，而是网格图，想必大家都会求从格点 K 到格点 D 的最短路——宽搜。这也是本题的正解算法。

f_i表示从起点出发，到达当前点 i 的最短路径数量；$step_i$表示到达当前点 i 所需的最少步数。点 i 能到达的点 j 有两种情况，走过和没走过。

没走过，根据宽搜性质，$step_i + 1$必然为到达点 j 所需的最少步数。f_j可以直接赋值为f_i。

走过，若$step_j == step_i + 1$，那么从起点出发到达点 i 的最短路径与 i 到 j 的边也能组成到达 j 的最短路径，f_j加上f_i即可。否则没有从起点经过 i 到达 j 的最短路，对f_j不做更改。

爱丽丝的寻路人偶显然经受住了考验（这可是融合了魔法与河童科技的产物！），不知你的智商是否也如此呢？

七色的人偶使

题目翻译：讲一个数拆成若干个质数之和，使得质数个数最少。

30 分+？分做法：
跑出 100 内所有质数，然后 dp 一下即可求解，是暴力部分分。
时间复杂度 O（K）

100 分做法：
哥德巴赫猜想：任何大于 2 的偶数都可以写成两个质数之和。
于是我们先看 K 是否是质数，是输出“1”。
然后若 K 是偶数，输出“2”。

对于 K 是奇数，我们考虑从中减去一个 3，得到偶数，故输出“3”。
但是奇数 K 可能能写成 2 + 一个质数，故需判断 K-2 是否是质数。

于是问题就来到了如何判断 N 是质数。

考虑 N=a*b，设 a≥ $\sqrt{n}$ 则 b 一定≤ $\sqrt{n}$，于是我们只判断 N 是否有因子 b，及

判断 N 能否有小于等于$\sqrt{n}$的因子。

时间复杂度 O（T*$\sqrt{K}$）

樱花结界

题目翻译：给你 N 个整数，要求从中选一些数字，使它们的总和为 K，求选择的方案数。

30 分做法：枚举每个数字选或不选，复杂度 $O(2^n)$。

100 分做法：首先对题目中的样例手动推导一下，样例一选择的方案为：
选 0 个，()。
选 1 个，(1, -1)。

总共 2 组方案，可以发现不选时的精彩程度为 0。

样例二选择的方案为
选 1 个，(10)。
选 2 个，(1, 9) (2, 8) (2, 8) (3, 7) (4, 6)。
选 3 个，(1, 2, 7) (1, 2, 7) (1, 3, 6) (1, 4, 5) (2, 2, 6) (2, 3, 5) (2, 3, 5)。
选 4 个，(1, 2, 2, 5) (1, 2, 3, 4) (1, 2, 3, 4)。

总共 11 种方案，重复数字也要参与计数。

简单的枚举会 TLE，不过a_i的数据范围也不允许使用桶之类的方法进行优化。这时，我们的人类智慧集合体——HXQ，提出了以二分解决这道题的思想。

现在，我将 H 神的圣谕记录于此：将 N 个数字分为两半，对左边和右边分别枚举，分别统计出能形成的数值。之后，将两边的数值进行组合，判断其和是否等于 K 即可。枚举的时间复杂度 $O(2^{n/2})$，组合最坏时间复杂度 $O(2^{n/2} * 2^{n/2})$。

追求极限效率的话，可以将能形成的数值重复值合并后排序，对于左半边数据从

小到大，对右半边数据从大到小枚举。

这样，组合后的和值显然具有单调性：和偏小，便继续对左半边数据从小到大枚举；和偏大，便继续对右半边数据从大到小枚举；和等于 K，由于已经去重，参与组合的数据无法再参与其它组合，左半边和右半边同时向内缩。组合最坏时间复杂度成功降为 O($2^{n/2}$)。

骚灵确实很吵闹，不过演奏出的曲子确实是悦耳的。虽说是要去往西行寺家赏樱，但是樱花是否能盛开还未可知呢~

白玉楼的庭师

题目翻译：从（0,0）出发只能向上，左，右。带不能走的禁止点，和步数，以及不能重复走，问多少种行走方案。
爆搜做法不在赘述。

最朴素的做法是 4 维的 dp，带方向的那种，那种整出定义后推一下转移就行了。
但是这里提供一个复杂点的做法，其优化思想以后可能会用到。
我们先不考虑禁止点的问题。

考虑 dp，定义状态$dp_{i,j,k}$为走到（i，j），剩余 K 步时，并且下一步**向上**的方案数

（就是这一行已经走完了左右方向）。

转移很简单$dp_{i,j,k}=dp_{x,j-1,k+1+|x-i|}$ （$x\in[-n，n]$）

但是分析可得这是 O（n^4）的 dp
于是我们想到要优化，很明显你只能优化转移时的求和。于是我们来观察一下求和的特征。

比如考虑$dp_{3,j,2}$ 的求和(表格里的是$dp_{i,j-1,k}$)：

k=5					
k=4					
k=3					
k=2					
k=1					
	i=1	i=2	i=3	i=4	i=5

很容易发现求和是两个一升一降斜线。
于是我们每次 dp 完第 j 行后，然后处理这一行所有这样斜线内的区间和。怎么

处理呢？

我们来看一下$dp_{3,j,2}$最后会贡献到哪些求和斜线当中

k=5					
k=4					
k=3					
k=2					
k=1					
	i=1	i=2	i=3	i=4	i=5

可以发现所有从蓝色方格引出的求和斜线都会包含$dp_{3,j,2}$。于是我们可以用差分处理掉前缀和。自然而然也就能得出求和斜线的区间和。

现在我们加上禁止点，同样考虑$dp_{3,j,2}$的求和，我们假想（4，j）有一个禁止点：

(表格里的仍然是$dp_{i,j-1,k}$的表，由于定义上 dp 下一步是向上走，故影响第 j-1 行求和斜线的反而是第 j 行的禁止点)

k=5					
k=4					
k=3					
k=2					
k=1					
	i=1	i=2	i=3	i=4	i=5

很容易发现求和斜线是不能越过禁止点的。
反之，每一个方格对求和斜线的贡献也不能越过禁止点。
于是只需要在差分上稍加处理即可。

时间复杂度 O（n^3），但是常数很大，所以真考场上推荐 4 维 dp

反魂蝶

虽然是第六题但是很水。

题目翻译：两个序列a_n，b_n ，找出一个子段[L,R] ($i\in[L,R]$)使得子段中所有a_i+b_i是 K 的倍数的 i 的个数与最小的a_i或 b_i的乘积最大。

我们先来看 K=1。
很容易发现这题变成了最大矩形面积，结合10^6的数据范围，我们可以猜测本题要单调栈。

考虑第一个因素：所有a_i+b_i是 K 的倍数的 i 的个数尽量大
翻译：区间越大越好

第二个因素：最小的a_i或 b_i尽量大

翻译：区间越小越好。

于是我们想到固定一个因素，这样剩下的因素是具有单调性的。
想到知道固定第二个因素（更容易固定啊）。

于是我们取一个 i 使得 $\min\{a_i,b_i\}$作为包含 i 的区间的最小值，剩下的我们只需要做到让这个区间尽量大，且没有比 $\min\{a_i,b_i\}$更小的a_i或 b_i存在。

于是：单调栈。
时间复杂度 O（n）