Knight：

不妨假设 x>=y>=0。

当 x<=2y 时，定义每一步的冗余值 wi=3-dx-dy，那么∑wi=∑(2-dx)=3*步数-x-y，显然我们只需要最小化冗余值。我们先只用(+2,+1)(若 x 为奇数则加一步(+1,+2))走到(x,y′)，然后通过将(+2,+1)替换为 2 个(+1,+2)使得 0<=y-y′<3。

若 y-y′=0，则冗余值为 0，显然最小。

若 y-y′=1，则将(+1,+2)替换为(+2,+1)和(-1,+2)或将 2 个(+2,+1)替换为(+1,+2),(+1,+2),(+2,-1)，冗余值为 2，显然最小。（此处需要特判(2,2)）

若 y-y′=2，则加上(+2,+1)和(-2,+1)，冗余值为 4，由于不存在冗余值为 1 的步，所以最小。

当 x>2y 时，定义每一步的冗余值 wi=2-dx，那么∑wi=∑(2-dx)=2*步数-x，显然我们只需要最小化冗余值。我们先只使用(+2,+1)走到(2y,y)，然后用(+2,+1)和(+2,-1)走到(x′,y)使得 0<=x-x′<4。

若 x-x′=0 则冗余值为 0，显然最小。

若 x-x′=1 则将之前的(+2,+1)改为(+1,+2)和(+2,-1)，冗余值为 1，显然最小。（此处需要特判(1,0)）

若 x-x′=2 则加上(+1,+2)和(+1,-2)，冗余值为 2，由 x/2+y 的奇偶性可知最小。

若 x-x′=3 则加上(+2,+1),(+2,+1),(-1,-2)，冗余值为 3，由 x/2+y 的奇偶性可知最小。

时间复杂度 O(t)

Tree：

f[i]表示 i 的子树中包含 i 的连通子集个数，两遍树形 dp 即可。注意 f[i]=0 时会无法进行除法，需要对子树求前/后缀积（或者记有多少个 0）。

时间复杂度 O(n)

Two Graphs：

考虑只有一张图时的费用流做法：将每个点和每条边建点，S 向每个点连边，每条边向 T 连边(有费用)。对于第 i 条边(a,b)，点 a、点 b 向边 i 连边。

有两张图时，将每条边拆成两个点 i1,i2，i1 向 i2 连边(有费用)，然后第一张图用之前的建图方式连到 i1，第二张图就 i2 向 a,b 连边。

时间复杂度 O(costflow(n1+n2+2m,n1+n2+5m))

Shopping：

显然可以将最贵的 min(m,凳子个数)个物品打折。

时间复杂度 O(tn)

Trophies：

反过来求 min1>max2 的方案数。

将数从大到小插入序列，每次统计当前这个数是 max2 的方案数。假设当前数的位置为 i，那么区间 2 的方案数是(i-l[i])*(r[i]-i)，l[i],r[i]分别表示上/下一个已经插入过的位置。区间 1 的方案数就是在 i 之前由已经插入的数组成的区间个数，用树状数组维护即可。

时间复杂度 O(nlogn)

Palindrome：

转化为求以下四个东西：

（1）对每个后缀求它和所有前缀的反串的 LCP 之和；

（2）对每个前缀求它和所有后缀的反串的 LCP 之和；

（3）对每个位置求以它为左端点的回文串数量；

（4）对每个位置求以它为右端点的回文串数量。

前两个用 SA+单调栈求，后两个用 manacher 求。

时间复杂度 O(nlogn)

stone：

首先，如果存在 a<=xi<=b，显然先手获胜。

否则，我们将 a~b 设为不可到达的状态，然后求其它状态的 sg 值。

当 a=1 时，从 b+1 开始，sg 值是 0~a+b-1 不断循环。

当 a>1 时，0~a-1 的 sg 值为 0，从 b 开始(假设 b 的 sg 值为 1)sg 值是 1 0 2 3 … 这样的循环，循环节长度为 a+b，每段长度为 a。

时间复杂度 O(tn)

travel：

把树分成 m 个连通块的方案数是 C(n-1,m-1)，乘上 m!就行了。

时间复杂度 O(∑n)

Metropolis：

把所有大都会设为源点跑多源最短路，记下每个点是由哪个源点扩展的。

如果从源点 i 出发走到了一个由另一个源点 j 扩展到的点 k，那么从 i 出发经过 k 的最短距离肯定是 dis[i][j]，那么就没有必要继续走下去了。所以只要枚举所有两端由不同源点扩展的边更新答案就行了。

时间复杂度 O((n+m)logn)

Graph Coloring I：

判一下是不是二分图就行了。

时间复杂度 O(n+m)

Graph Coloring II：

由于要求删掉奇环以后图连通，我们考虑先找出一些边使得图连通（当然是生成树啦），然后剩下的边如果不是二分图就可以找个奇环删掉。

一开始我们随便选个点染黑，然后将它的所有未染色的相邻点染白。接着我们每次找一个和白点相邻的未染色点，重复以上操作。这样我们得到了一棵黑白相间的树，并且黑点直接没有任何边。那么如果剩下的边是二分图，将白点分成两种颜色即可。

时间复杂度 O(n+m)