城堡保卫战

算法一

对于测试点 1，n=1，只需判断 h1 的生命值是否能承受唯一怪物的伤害，即h1>a1b1 即可。期望得分 5 分。

算法二

使用搜索算法。DFS 枚举每次打哪个怪物，记录当前已经打败了哪些怪物和剩余生命值。时间复杂度 O(n!m)，可以通过前 6 个测试点获得 30 分。如果不会写 DFS，可以手动枚举所有的排列情况通过 n≤3 的 15 分。

算法三

显然这个搜索是可以记忆化的，考虑使用状态压缩 DP。定义 f(S,h) 表示已消灭怪物集合为 S，剩余生命值为 h 时，最多还能击败多少怪物，则$f(S,h)=max\{f(S\cup\{i\},h-(a_i+d|S|)(b_i+d|S|)\}$, 其 中$i\notin S$，$h-(a_i+d|S|)(b_i+d|S|)>0$。当不存在这样的$i$时$f(S,h)=0$。做完 DP 之后，对于每个猎人$j$,可以直接查询其最多能消灭的怪物数$f(\emptyset,h_j)$。时间复杂度$O(2n*n\cdot maxhi+m)$

，期望得分 40~45 分。

算法四

上述状压 DP 还可以进一步优化。你会发现 h这一维范围很大，然而 f(S,h) 范围很小，不妨把两者交换一下！定义 $f(S,k)$ 表示已消灭怪物集合为 S，之后消灭 k 个怪物至少需要损失的生命，则$f(S,k)=max_{i\notin S}f(S\cup i,k-1)+(a_i+d|S|)(b_i+d|S|)$

这样查询猎人 j 能消灭的怪物数就是求最大的 k 使得 $f(\emptyset,k)<h_i$。时间复杂度 $O(2nn+mn)$，期望得分 50~55 分。

算法五

对于测试点 16,17，d=0 ，也就是消灭怪物是顺序无关的。由于消灭怪物 i 消耗的生命值为 a_ib_i，我们将怪物按照 a_ib_i 从小到大排序，那么对于每个猎人，如果能消灭 k 个怪物，那么一定是贪心取 a_ib_i 最小的 k 个怪物。排序后预处理 a_ib_i 的前缀和，对于每个猎人，可以使用二分法求出其最多能消灭的怪物数量。时间复杂度 $O((n+m)logn)$，期望得分 15 分。结合上述算法期望得分 60~65 分。

算法六

$d=0$ 的数据可以贪心，那么 $d\neq 0$ 的数据是否也有贪心性质呢?

假设 i,j 是猎人的打怪顺序中两个相邻的怪物，i 在 j 前面，不妨设在打到 i 之前，i 的攻击力和防御力为 a_i,b_i，j 的攻击力和防御力为 a_j,b_j，则消灭 i,j 总共损失生命$h=a_ib_i+(a_j+d)(b_j+d)$

如果交换i,j 顺序，即先消灭j，再消灭 i，那么总共损失生命$h\prime=a_jb_j+(a_i+d)(b_i+d)$,由于
$h\prime-h=(a_i+b_i)d-(a_j+b_j)d$，当 $a_i+b_i\leq a_j+b_j$ 时，$h\prime\leq h$，也就是交换 i,j 顺序更优。

因此我们有如下结论：一定存在最优解，使得怪物按照 a_i+b_i从大到小的顺序消灭。这样，将怪物按照 a_i+b_i 从大到小排序，那么答案序列是这样排序后的一个子序列。用 DFS 枚举子序列，并更新答案。复杂度 $O(2nm)$，期望得分 55~60 分，结合算法五期望得分 65~70 分。

算法七

上述算法的 DFS 枚举子序列同样是可以记忆化的。将怪物按照 a_i+b_i 从大到小排序后，

考虑 DP，记 $f(i,j,h)$ 为已经确定前 i个怪物是否消灭，前 i个怪物消灭了 j个，剩余生命 h，之后最多能消灭多少个怪物，则

$$f(i,j,h)=\begin{cases}=0 & i>n\\ =f(i+1,j,h) & h-(ai+1+dj)(bi+1+dj)\leq 0\\ =max\{f(i+1,j,h),f(i+1,j+1,h-(a_i+1+d_j)(b_i+1+d_j)+1)\} & otherwise\end{cases}$$

DP 之后，每个猎人 j 最多消灭怪物数可以直接查询 $f(0,0,h_j)$。时间复杂度 $O(n^2max\{h_i\}+m)$，期望得分 55 分，结合算法五、六期望得分 80 分。

算法八

最后，仿照算法四对算法三的优化，将上述 DP 进行优化就是正解了。

同样将怪物按照a_i+b_i 从大到小排序，然后记 $f(i,j)$ 为前 i 个怪物消灭 j 个，至少消耗多少生命值，则

$$f(i,j)=\begin{cases}0 & \text{j=0}\\ +\infty & \text{j>i}\\ min\{f(i-1,j),f(i-1,j-1)+[a_i+d_{j-1})][b_i+d_{j-1})]\} & 0<j\leq i\end{cases}$$

DP 之后，对每个猎人 j 可以使用二分法求出其最多消灭的怪物数量。

最后我们成功地解决了这个问题，时间复杂度 $O(n^2 + mlogn)$，期望得分 100分。

树维

根据期望的线性性，可以转化为讨论每个点出现在虚树上的概率。出现在虚树上要满足要么他本身被选中，要么他至少两个子树中都有点被选中。

第一种显然是p_i 。

第二种可以通过计算他的反面来得到。也就是要么他只有一个子树中有点被选中，要么他没有任何一个子树中的点被选中。

那么可以通过预处理子树中一个点都没选中的概率来计算。

野外旅游

我们使用分治法解决，对于当前分治区间【L, R】,对于$m = [(L + R)/2]$，我们考虑所有过 m 的区间，我们以 m为根做 dfs 预处理，接着对于每个点 x 统计从 x 到根的路径上点的编号的最大值和最小值，令Q_i表示区间 $[i, m] or [m, i]$ 的区间的点到根的路径上权值最大值，令P_i表示其最小值。显然一个区间 $[i, j]$ 可行的必要条件是$Pi \geq i, Qj <= j$，我们只考虑 i, j,这样的说i, j，对这样的i, j，他们还需要满足$Q_i <= j, P_j >= i$，可以发现随着 i 的减小，满足条件的最小的j和最大的 j 都是单调不降的，所以可以使用$Two - pointersO(n)$解决.于是总的分治复杂度就是$O(nlogn)$了。

路在何方

方法一

暴力搜索，期望得分 10 分。

方法二

考虑将这个问题想象成一个博弈问题，你每走到一个点你的对手都可以决定堵上一条与该点相邻的边或者不堵，考虑设 $dp[i]${ 表示i号节点的答案。建出原图以 v为根的最短路树，考虑对手要堵一定会堵上该点到其最短路树上的父亲的边，于是对于每一个点 x，求出堵住了他到最短路树上的父亲的边后到 v的最短路，设其为 $val[x]$。考虑由于最优决策一定是不会成环的，所以之前得到的一些边没有堵住的信息不会影响当前点的决策，于是就可以得到这样的转移:
$dp[i] = max(val[i], min\ dp[j] + e(i, j))$
将 max 和 min 换-下位置可以得到:
$dp[i] = min_{(i,j)\in e} max(val[i], dp[j] + e(i, j))$
于是求出来val后直接类似 $diikstra$ 做一下即可。考虑 val怎么求，暴力每次断一条边求最短路，时间复杂度 $O(nmlogn)$，期望得分 40 分。

方法三

考虑优化求 val的过程，容易发现只会走一条非树边，假设当前点为x，走的非树边为(y, z, x)，那么走过的距离为 $dep[y] + dep[z] + w - dep[x]$，且必须满足: x在最短路树上是在 y到 z的路径上，且不是 $lca(y, z)$。于是考虑对于每条非树边 (y, z, w)，将断掉 y到 z上的路径内的边都跟 $dep[y] + dep[z] + w$最小值。最后再将每条边的值减去底部点的深度即可。如果采用树剖/线段树实现的话，复杂度/常数较高，只能得到 70 分。

方法四

考虑将所有边的权值按照从小到大排序，这样我们每次就只需要维护对于一个路径上的边，如果它之前没有被覆盖，就将它覆盖成某一个值的操作，这个可以之间通过树上并查集实现。
时间复杂度$O((n + m)logn)$，期望得分 100 分。