T1

首先可以把透明的部分去掉，假设有n'行不透明的格子和m'行不透明的格子。

原题可以转化为一个$n' \times m'$的箱子，每行每列至少有一个不透明的格子，求其方案数。

小数据可以状压。

考虑容斥。

对于70%的数据，枚举至多覆盖多少行和多少列，具体式子就是
$\sum_{i=0}^{n'}\sum_{j=0}^{m'}(-1)^{n-i+m-j}*C_{n'}^{i}C_{m'}^{j}*2^{i*j}$

对于100%的数据，发现我们在枚举行的同时，列同样可以一起容斥掉，枚举至多覆盖多少行，假设是i行，则有C_n^i种选择，对于剩下的m列，每一列都有2^i-1种方案（不能为空），则方案数为$C_n^i*(2^i-1)^m$，具体式子就是$\sum_{i=0}C_n^i*(2^i-1)^m$

T2

小数据可以直接暴力枚举或模拟实现。

关于本题，我们可以发现如下一个结论：

所有点至多被覆盖两次

于是就可以dp，把所有小区间按照右端点从小到大排序，设计$dp[i][j]$表示放到了第i个区间，$[j,r_i]$被覆盖了一次。转移分两种：

- $r_j+1==l_i$：所以$[l_i,r_i]$全都没被覆盖，$dp[j][1\sim r_j]$直接跟w_i取max
- $r_j\in[l_i,r_i)$有一段区间被覆盖，$dp[j][1\sim l_i]$跟w_i+w_j取max

直接转移是$O(n^2m)$的，套上一个前缀最小值优化可以做到$O(n^2+nm)$。

T3

10pts

暴力枚举每一条边的墙高度，然后暴力bfs染色然后暴力check。

50pts

正难则反，考虑计算所有房子都危险的方案数，最后除以总方案数$\prod_i(r_i-l_i+1)$即为其概率。

定义$f[i][j][0]$表示i这个子树所有点已经被走完，且能到根的最强僵尸能力为j（默认离散）。

定义$f[i][j][1]$表示i这个子树没有被走完，需要的外援能力至少为j。

考虑树形dp合并两个节点:u为儿子p为父亲。

然后我们枚举边权v，然后分类讨论： 对于$f[u][a][0]$和$f[p][b][0]$，若b走得过边v，那么就更新$f[p][max(a,b)][0]$，否则就更新$f[p][a][0]$

对于$f[u][a][0]$和$f[p][b][1]$，若a走得过边v，且$a\geq b$，那么就更新$f[p][a][0]$，否则更新$f[p][b][1]$

对于$f[u][a][1]$和$f[p][b][0]$，若b走得过边v，且$b\geq a$，那么就更新$f[p][b][0]$，否则就更新$f[p][max(a,find(v))][1]$，其中$find(v)$表示第一个比v大的僵尸能力

对于$f[u][a][1]$和$f[p][b][1]$，更新$f[p][max(max(a,b),find(v))][1]$即可。

你可以把边权离散了，或者是不讨论v，直接枚举m个僵尸，然后判断和边权大小即可。

这样是$O(n \times m^3)$或者$O(n^2m^2)$。~~（其实可以有70）~~

70pts

上述方法中可以使用前缀和优化或者是差分或者使用填表分类能消去一个n或者m。

~~其实前缀和优化能变成$O(n^2)$的。~~

100pts

首先僵尸能力升序排，以$dp[u][k]$表示u这棵子树被侵占了，且u子树中能到达u这个结点的最强僵尸编号为k。

考虑如何转移：

现在子树是u，假设v是u的儿子，假设$v \to u$这条边被k这个僵尸通过的方案为x_k，通不过的方案为y_k，分三种情况转移（注意区分以下：

- v被k侵占了，$dp[u][k] \cdot dp[v][k] \cdot x_k \to dp[u][k]$
- v被其子树内比k强的僵尸侵占了，为了使得干掉u的僵尸能力不超过k，要阻碍v子树上的僵尸过去，$dp[u][k] \cdot \sum_{t>k} y_t \cdot dp[v][t]$
- v被其子树内比k弱的僵尸侵占了，为了使得干掉v的僵尸能力弱于k，要阻碍k过去，$dp[u][k] \cdot y_k \cdot \sum_{t<k} dp[v][t] \to dp[u][k]$

注意考虑这棵子树里是否有这只僵尸再进行转移。

直接转移复杂度是$O(nm^2)$的，前后缀优化一下可以做到$O(nm)$。