A. 向量

每次操作如果 $a+b<10$ ，数位和不变且序列长度减少 1 。若 $a+b\geq 10$ ，数位和减少 9 而序列长度不变。故答案为 $[\frac{\sum S_i-1}{9}]+|S|-1$

B. 顺序

最大的 t 是 $nm-1$ 很好构造，构建一颗S型或者螺旋的生成树都可以。

最小可行的 t 不小于 $n+m-2$ ，因为这是对角两点的距离。可以发现 n,m 是偶数的时候无法达到这个下界，因为角上的四个点总有两点在生成树上距离大于 $n+m-2$ 。因此在 $2|n,2|m$ 时下界多 1 。下界可以构造一个“王”字形的图来得到，即在最靠近中间的地方作一条竖线，然后向左右连边。（或者旋转 90° ）

剩下介于下界和上界中间的 t ，由下界的生成树两端如S形扩展即可。

C

考虑对于表达式的一个子部分，你只关心它的值是 0 还是 1。

建出表达式树后，记 $f_{i,0/1}$ 为子树 i 的值为 $0/1$ 的方案数，转移时可以枚举所有字符的填法和所有数字的填法，然后讨论一下转移。

记表达式的左右两个子树的部分为 j 和 k

当运算符为 `^` 的时候，$f_{i,1}=(f_{j,0}\times f_{k,1}+f_{j,1}\times f_{k,0}),f_{i,0}=(f_{j,1}\times f_{k,1}+f_{j,0}\times f_{k,0})$ 。

当运算符为 `|` 的时候，$f_{i,1}=(f_{j,1}\times f_{k,1}+f_{j,0}\times f_{k,1}+f_{j,1}\times f_{k,0}),f_{i,0}=(f_{j,0}\times f_{k,0})$ 。

当运算符为 `&` 的时候，$f_{i,1}=(f_{j,1}\times f_{k,1}),f_{i,0}=(f_{j,0}\times f_{k,0}+f_{j,0}\times f_{k,1}+f_{j,1}\times f_{k,0})$ 。

当运算符为 `?` 的时候，需要把所有运算符的转移情况加起来。

D. 熵值

在两点之间操作后，全局的点权和会少 $d=\sqrt{(x_u-x_v)^2+(y_u-y_v)^2}$ 。显然在更优的操作下两点间只会操作一次，进一步地，操作会形成一种（无向的）树形结构。同一个连通块 $|V|$ 内点权最大值为 $\frac{\sum w-\min\{\sum d\}}{|V|}$ ， $\min\{\sum d\}$ 即是最小生成树的边权和，这个上界容易证明是一定可以达到的（每次先让叶子达到这个值，然后去掉叶子）。不同连通块之间明显互不影响。

因此先求出不同连通块的点权，再dp原图的连通块划分即可。