纯口胡的 Solution:

T1:

对于每个岛屿求出其[l,r]的区间，然后问题转化为选取最少的点覆盖所有的区间；
那么我们把区间按照右端点排序，记 last 为上一个选取的点所在的区间，然后当 q[i].l>q[last].r 的时候新建一个点即可；

T2:

经典的合并果子问题，用二叉 Huffman 树实现即可；

T3:

给定一个无向图，可以把 k 条边的边权缩为 0，然后询问 1-n 路径上的最大边权的最小值；
做法比较多；
1.因为 n<=1000,k<1000,那么我们设 dis[i][j],表示到 i 号节点用了 j 次免费机会的最大值；
用 spfa 来实现这个 dp 转移即可；
2.一种更为优秀的算法，因为答案满足可二分性，所以我们可以二分这个最大值 Mid，然后我们定义 dis[i]表示到 i 号节点经过的边权>Mid 的最少的边的个数，判断是否 dis[n]<=k 来进行 check;

T4:

有 n 个带权区间，要求选取的区间不相交且两个区间的距离>=R,问能获得的最大权值；
首先把区间按照右端点排序，设 dp[i]表示到第 i 个区间的最大收益，转移也很 simple:
dp[i]=max(dp[i],dp[j]+val[i]) [l[j]+R<=r[i]];
dp[i]=max(dp[i],dp[i-1]);
转移可以用数据结构优化，但没必要了；

T5:

把一个序列变为不降或不升子序列的最少代价，定义每个数改变的代价为 abs();
先考虑变为不降序列的做法，不升则类似：
n<=1000，那么我们考 n^2 dp，设 dp[i][j],表示把 i 位置上的数改为 j 并且保证合法的最小代价；
因为 j 很大，我们把第二维离散化即可，显然转移只会和前面一个有关；
朴素的转移：
dp[i][j]=min{abs(hsh[v[i]]-hsh[v[j]])+dp[i-1][k] (k<=j)};
显然我们需要维护 dp[i-1][k] (k<=j) 的最小值就可以完成优化了，那么我们用一个数组维护前缀最小值即可，设为 Min[i][j];
那么转移变为
dp[i][j]=abs(hsh[v[i]]-hsh[v[j]])+Min[i-1][j];
Min[i][j]=min(Min[i][j-1],dp[i][j]);
这两个数组都可以滚，然后不升就是维护后缀最小值即可；

T7:
问用 n 个数，每个数无限个，能凑出 k 的方案数；
相当于用完全背包累计方案，f[i]+=f[i-a[j]]，但是 f 数组可能有是高精度数，但是位数还是比较少，可以用两个 long long 的数来模拟高精度进位，操作和 9.11 的考试 T3 惊人的类似，我们用一个 long long 存后 18 位记为 a，另一个 long long 存前 18 位记为 b，那么(b1*long long+a1) + (b2*long long+a2) 就相当于是(b1+b2+(a1+a2)/long long)*long long +(a1+a2)%long long;然后分别输出 b，a 即可；

T8:

分组背包，问能凑出多少数字；
就是正常的分组背包问题，解法经典略过，二进制分组会 TLE，注意 c[i]*a[i]>=m 就用完全背包;

T9:

n 个点，每个点有两个权值 a,b,在其中选点，问满足∑a>=0,∑b>=0 的条件下，∑a+∑b 的最大值；

每个点只能选一次，我们考虑用 0/1 背包来实现，那么我们把 a 当做重量，b 当做权值，那么相当于是满足重量和权值均>=0 的条件下，重量+权值的最大值，我们用 dp[i]表示容量为 i 的最大权值，让后按正常的 0/1 背包进行转移即可，注意可能有负数，所以空间×2 即可；

T10:

n 个奶牛叠罗汉，然后每个奶牛有重量和力量，需要求出一种排列，使得每个奶牛，在他之上的奶牛的重量-他的力量的最大值最小；

直观上感觉，需要重量大和力量大的在下面，做法是按照 s+w 进行排序，然后输出答案；
具体思路和证明和国王游戏类似,即证明交换后不会更优;
设 Di 表示第 i 头奶牛的难受值，Wi 表示第 i 头奶牛的体重，Si 表示第 i 头奶牛的力量，
令 i,j 相邻,且 Wi+Si>Wj+Sj，设∑表示 i 和 j 上面的奶牛的重量之和
当 i 在 j 的上方时有
Di=∑−Si ①
Dj=∑+Wi−Sj ②
当 j 在 i 的上方时有
Di=∑+Wj−Si ③
Dj=∑−Sj ④
显然我们可以得到
③>①,②>④,②>③
这里面②最大，所以如果我们让 i 在 j 的上方最终答案一定不会更优，即证得此贪心策略的正确性。

T11:

给定一个字符串，可以增删字符，每个字符增删的代价不同，问把其变为一个回文串的最小代价；
很容易想到区间 DP 的模型，定义 dp[i][j]表示把区间 i-j 变为回文串的最小代价，每次只会在区间两端进行操作，那么主要说一下 s[i]!=s[j]的转移：
dp[i][j]=min(dp[i][j],dp[i][j-1]+min(d[s[j]],a[s[i]]);
dp[i][j]=min(dp[i][j],dp[i+1][j]+min(d[s[i]],a[s[i]]);

T12:

题意就是 n<=10，所以爆搜；
用状压 Spfa 实现即可

T13:

Vijos 遭遇战，当然也可以用线段树优化 Dp 来实现；
只说明 Spfa 的做法，问题变为 1—n 的最短路，然后从 si 向 ti+1 连边，每天往回连边即可；

T14:

网格图上有 n 个点，要求选择一条至少涵盖三个点的等差路径，使得其覆盖的点最多；

O(n^2)的枚举起点和第二个点，然后 O(n)的计算答案，有一个剪枝，就是判断起点能否跳进来，这样保证了路径只会在其最优的起点被枚举一次；