solution

math

发现对于任意一个加号总存在一个序列只有这一位是个减号从而抵消贡献，所以最后有贡献的只能是所有的前缀积，对于一个长度为i的前缀积，贡献为$numi * 2 * 3^{n-i-1}$,直接用线段树维护就行了。

bf

我们设Fkb所选的点集合为A,博弈doe所选的点集合为B，全集为Q，那么最后的答案就是

$$v = \sum_{pi,pj\in A,i<j} dis(pi,pj) - \sum_{pi,pj\in B,i<j} dis(pi,pj)$$

$$= \{\sum_{pi,pj\in A,i<j} dis(pi,pj) + \sum_{pi,pj\in B,i<j} dis(pi,pj) + \sum_{pi\in A,pj\in B} dis(pi,pj)\} -$$

$$\{\sum_{pi\in A,pj\in B} dis(pi,pj) + \sum_{pi,pj\in B,i<j} dis(pi,pj) + \sum_{pi,pj\in B,i>j} dis(pi,pj)\}$$

$$= \sum_{pi,pj\in Q,i<j} dis(pi,pj) - \sum_{pi\in Q} \sum_{pj\in B} dis(pi,pj)$$

Fkb要最大化V,而博弈doe要最小化V,算是前半部分是定值，而后半部分只与B集合有关，我们把每个点的就是他到所有点的距离之和，所以我们排个序，两人轮流取贡献最大的点。

bag

设$dp[i][j]$表示选了i个物品，体积为j的方案数，转移时枚举当前物品用了几个，这样你就得到了一个效率接近$O(n^3)$的dp,如果你发现了大于$\sqrt{n}$的物品是用不完的，所以直接对这些物品做完全背包，对剩下的物品做和刚才一样的dp，时间复杂度可以降为$O(n^2)$.

仔细思考这两个n^2的dp，对于完全背包的dp，我们这样改进，$dp[i][j]$表示选了i个物品，体积为j的方案数，转移时，我们有两种抉择，一种是加入一个体积为$\sqrt{n}+1$的物品，$dp[i+1][j] += dp[i][j-\sqrt{n}-1]$,另一种是将所有物品体积都加一，$dp[i][j+i] += dp[i][j]$.这样时间复杂度便降为$n\sqrt{n}$，对于多重背包的$dp$，我们设$dp[i][j]$表示前i种物品体积为$j$的方案数，但我们不枚举用了几个，事实上$dp[i][j] = \sum_{k<=i} dp[i-1][j-i*k]$，记个类似前缀和的东西转移就好了，复杂度也是$n\sqrt{n}$的。