糖果

Algorithm 1

直接 2^{2n} 枚举每个糖果买或不买，并更新答案。

时间复杂度：$\Theta(2^{2n} \cdot n)$，期望得分：30 分。

Algorithm 2

如果我们确定了买的糖果数量，那么我们就要最大化两袋糖果价值和的最小值。因此我们在每家商店一定都是选择愉悦度尽可能大的糖果，于是枚举在两家商店分别买了多少糖果，然后计算答案即可。

时间复杂度：$\Theta(n^2)$，期望得分：90 分。

Algorithm 3

不妨设是第一家商店买的糖果愉悦度之和最小（如果是第二家，那么把两家糖果交换一下再做一遍即可）。我们枚举第一家商店买了多少糖果，那么此时这个买糖果方案的愉悦度就已经确定了，所以我们要最小化花费即可。于是在第二家买糖果一定是购买最少的糖果，使得愉悦度之和不小于第一家买的糖果。可以二分，也可以通过单调性然后 $Two - pointers$ 求出这个位置。

时间复杂度：$\Theta(n)$ 或 $\Theta(n\log_2 n)$，期望得分：100 分。

排序

Algorithm 1

回顾选择排序，选择排序是每次把未排序部分中，最小的换到头部。那么我们类似地每次求出最小的，然后翻转这个前缀即可（因为后面的部分都是未排序的，所以翻转并不会对正确性造成影响）。

时间复杂度：$\Theta(n^2)$，操作代价：$\Theta(n^2)$，期望得分：60 分。

PS：注意这里如果翻转过去再回来来实现交换的话，那么会因为常数太大而无法通过 $n = 5000$ 的测试点。出题人

在第 12 个测试点卡了这个常数（第 11 个点放过去了）。

PS：这样做也能直接通过第 13 个测试点，因为 0/1 序列每次第一个 0/1 都不会太远，均摊下来常数很小。

Algorithm 2

回顾归并排序，归并排序是每次把左右两部分都排好序，然后再把他们合起来。那么对于 0/1 序列而言，排好序等价于前面都是 0，而后面都是 1。那么我们每次把左半边的 1 和右半边的 0 进行翻转，即可把整段都排好序。

时间复杂度：$\Theta(n\log_2 n)$，操作代价：$T(n) = 2T(\frac{n}{2}) + \Theta(n)$，解得 $T(n) = \Theta(n\log_2 n)$，期望得分：50 分（即 0/1 序列的点）。

Algorithm 3

回顾快速排序，快速排序是每次选择一个 *pivot*，然后把 $\le pivot$ 的放到左边，$> pivot$ 的放到右边，然后递归下去。那么如果我们把 $\le pivot$ 的看做 0，$> pivot$ 的看做 1，我们就相当于要把这个 0/1 序列排序。直接套用 `Algorithm 2` 即可。

时间复杂度：$T(n) = 2T(\frac{n}{2}) + \Theta(n\log_2 n)$，解得 $T(n) = \Theta(n\log_2^2 n)$，操作代价也是 $T(n) = \Theta(n\log_2^2 n)$，期望得分：100 分。

最小生成树

Algorithm 1

直接阶乘枚举图中每条边的边权，然后跑最小生成树即可。

时间复杂度：$\Theta\left(\left(\frac{n(n-1)}{2}\right)! \cdot n^2\right)$，期望得分：20 分。

Algorithm 2

考虑 $a_i = i$ 的点，我们先确定这 $n-1$ 条边的形态，然后其他边都可以任意加。因为 n 个点带标号的树的个数是 n^{n-2}（*Matrix Tree* 定理，直接打表应该也能发现这个性质），所以我们的答案就是 $n^{n-1} \cdot (n-1)! \cdot \left(\frac{n(n-1)}{2} - (n-1)\right)!$，直接计算即可。

时间复杂度：$\Theta(n^2)$，期望得分：40 分（即 $a_i = i$ 的点）。

Algorithm 3

回顾 *Kruskal* 算法求最小生成树的过程，是按照从小到大的顺序加入每一条边，如果这条边的两个端点不在同一个连通块中，我们就加进去，否则不加。

那么我们模仿这个过程，考虑动态规划：也是按照从小到大的顺序考虑每条边，然后状态记录当前的连通块情况，转移直接枚举连接哪两个连通块即可。记录连通块情况可以记录每个点所在的连通块中，编号最小的点。这样状态数不太多，应该可以通过 $n = 20$ 左右的测试点。

时间复杂度：$\Theta(\text{状态数} \cdot n^2)$，期望得分：60 分左右。

Algorithm 3

可以发现我们在转移时，并不关心每个连通块里面具体有哪些点，而是只要知道每个连通块的大小。于是我们改为记录每个大小的连通块有多少个，这样的状态数是把 n 划分成若干无序正整数的方案数，记作 P_n。P_n 不太大，P_{40} 也只有 37338。

时间复杂度：$\Theta(P_n \cdot n^3 \log_2 n)$（多一个 $\Theta(n\log_2 n)$ 是因为合并两个连通块之后，要排序得到新状态），期望得分：80 分左右。

Algorithm 4

我们在转移时，只关心连通块大小，所以我们可以改为枚举合并的两个连通块的大小。因为和为 n 的若干个数只可能有 $\sqrt{n}$ 个不同的取值，所以枚举两个连通块的复杂度变为 $\Theta(n)$。

时间复杂度：$\Theta(P_n \cdot n^2 \log_2 n)$，期望得分：100 分。