CodeM2018 资格赛题解

Order：

Idea：

这是一个在点外卖的时候经常会遇到的问题，究竟要不要因为特价商品而放弃满减呢？这题的思路是要选择特价商品的话，就选所有的特价商品，否则就走满减。所以只要计算一下选择八折后的最便宜的价格，以及不选择八折的最便宜的价格，比较一下即可。

时间复杂度为 $\mathrm{O}(\mathrm{n}+\mathrm{m})$。

Solution：

https://paste.ubuntu.com/p/NrTk4pM4qx/

Cola：

Idea：

考虑每种可乐带来的期望收益，每种可乐会有 $\frac{m}{n}$ 的概率被小美喝，$\frac{m-n}{n}$ 的概率被小团喝，所以期望收益为$\frac{m}{n}\times a+\frac{m-n}{n}\times b$。因为期望的线性可加性，我们只要全买期望收益最大的可乐就可以了。

时间复杂度为 $\mathrm{O}(\mathrm{k})$。

Solution：

https://paste.ubuntu.com/p/jQMKYd8X4J/

Worldcup：

Idea：

假设 16 进 8 为第一轮比赛，8 进 4 为第二轮比赛，4 进 2 为第三轮比赛，决赛为第四轮比赛，那么我们可以用 f[i][j]表示在第 i 轮比赛后，j 还没有被淘汰的概率，对于某场特定的比赛，我们枚举一下参赛双方，就知道了每个人晋级下一轮的概率。活到最后的就是赢家。

时间复杂度为 $\mathrm{O}(1)$。

Solution：

https://paste.ubuntu.com/p/m2Wx5N7pVV/

Score：

Idea：

这是一道考验参赛者是否细心的题。我们只要枚举丢失的那个成绩的值，然后计算出每个人的成绩，按照题意看一下是否有可能入选即可。需要注意的是这道题对于精度的要求较高。

时间复杂度为 $\mathrm{O}(\mathrm{nmC}+\mathrm{nC}\log n)$。

Solution：

https://paste.ubuntu.com/p/ZzKxPBc5nG/

City：

Idea：

首先，我们用$E = \{(i,j)\}$，表示所有航线/车次的集合。$s(i,j), t(i,j), cost(i,j)$表示该航线/车次的出发时间、到达时间和花费。

设$f[i]$表示从城市 i 出发，可以到达 n 号城市的最晚航班/车次的出发时间（不考虑错过班次），$dist[i][t]$表示在 t 时刻到达城市 i 且随时可以起飞的最小花费（考虑错过班次）。

$f[i]$的转移式如下

$$f[i] = \max\{s(i,j) | (i,j) \in E, t(i,j) < f[j]\}$$

于是，我们可以将所有的边(i,j)反向，从 n 号城市向前转移。由于转移具有后效性，使用 spfa 算法进行转移。

对于$dist[i][t]$，若$f[i] \le t$且$i \ne n$，表示 t 时刻在 i 号城市已经没有可以换乘的备选航班/车次了。此时$dist[i][t]$无法进行转移。

否则，有可选转移

$$\mathrm{dist}[i][s(i,j)+1] \to dist[j][t(i,j)+1] + cost(i,j)$$

$$\mathrm{dist}[i][t] \to dist[i][t+1]$$

即飞到下一个城市或者在原城市等待下一个航班/车次。

$dist[i][t]$的转移方程构成了一张有向无环图，我们直接在 DAG 上做 DP 即可。

时间复杂度为$O(48n + m)$。

经过细致的实现，复杂度也可与 24（一天的小时数）无关（即将每个点的 t 按相关的边离散化，此题不考察）。

Solution：

https://paste.ubuntu.com/p/yjGrqkpKgK/

Match：

Idea：

可以看到这是一个匹配模型，先建个超级源点 S 和超级汇点 T，使得它变成一个带权网络流模型。

相当于要对每个$i \in [1,n]$，求一个流量为 i 的，最大权最小的 S 到 T 的流。

考虑最暴力的做法：每次二分最大权的值，然后就变成了普通的网络流问题，时间复杂度为$O(n^2 m \log m)$。

考虑以上暴力，可以发现 i+1 的方案可以由 i 的方案增广 1 的流量得到，于是可以二分个答案，然后看看从 S 到 T 能不能有新的增广路。时间复杂度$O(nm \log m)$。

现在主要的瓶颈是二分，考虑我们的问题其实是：给一个有向图，和一个边集 E，要在 E 中选一些边加入有向图，使得 S 能到达 T，且选的边的最大权尽量小。

我们可以将边从小到大排序，然后一条条加进去，动态维护 S 是否可以到达 T。

维护一个集合 G，表示$\forall x \in G$，S 能到达 x。

每次加入一条边(x,y)时，如果$x \in G$且$y \notin G$，那么令$G = G + \{y\}$。

每次新加入一个点到 G 中时，检查一下这个点的所有出边，一直加点进去。

由于每个点只会被加入 G 一次，所以每条边也只会被枚举一次，所有总的时间复杂度是$O(m)$。

于是得到了时间复杂度为$O(nm)$的做法。

Solution：

https://paste.ubuntu.com/p/DbPmrz6CMn/