# Candy

【题目大意】

有N颗糖果和M个小孩，老师现在要把这N颗糖分给这M个小孩。每个小孩i对每颗糖j都有一个偏爱度Aij，如果他喜欢这颗糖，Aij = k，否则Aij = 1。小孩i觉得高兴当且仅当ΣCij×Aij >= Bi，j=1,2,…,N，若他分得了糖j，Cij = 1，否则Cij = 0。问能否合理分配这N颗糖，使得每个小孩都觉得高兴。

【建模方法】

（最大费用最大流）

本题有一个突破点就是：他喜欢这颗糖，Aij = k，否则Aij = 1，关键在于如果他不喜欢这个糖分给他一样可以获得1点的欢乐值。也就是说如果之前分配了的糖给了小孩一定的欢乐值，不够bi，可以直接用随意的糖去填满。

首先我们要求欢喜值>=bi，是否可以认为当我获得欢喜值为bi后，多余欢喜值对这个结果的满足是没有贡献的。也就是说，你可以用一个容量来控制分配糖对小孩欢喜值的控制，让获得欢喜值最多为bi。如果不够，最后用一些1的糖来填满。

而这个容量就是bi/c，获取贡献为k，bi%c（>1）的也是可以用一个能让这个小孩欢喜的糖来贡献，但是其费用只为bi%c。

对于小孩来说，最大费用最大流后，糖分配的贡献值为最大费用，剩余糖就是（n-最大流），然后用这些糖去填不满的，只要满足总和大于Σbj。就可以分配了。

具体建模方案1：

(s,i,1,0);

(i,j,1,0);

(j,t,bj/k,k);

If(bj%k>1)

(j,t,1,bj%k)

Ans = 费用 + N – 最大流 >= Σbj 则满足要求

具体建模方案2：

(s,I,1,0)

(I,j,1,0)

(j,t,bj/k,k-1);

If(bj%k>1)

(j,t,1,bj%k-1);

Ans = 费用+N >= Σbj 则满足要求

当k = 2时候，可以不用费用流，最大流即可。

一种最直观的想法就是每颗糖i作为一个点并连边(s, i, ?)，每个小孩j作为一个点并连边(j, t, Bj)。若小孩j喜欢糖果i则连边(i, j, 2)，否则连边(i, j, 1)，然后求一次最大流看是否等于ΣBj。但是问题也很明显，我们还没有给与源点关联的边确定容量。实际上我们无法确定它们的容量，因为最大流无法实现这样一种控制：一个点有若干条出边，容量不尽相同，现在要求经过该点的流可以任选一条出边流走，且一旦选定之后就只能从这条边流而不能再进入其他的出边。因此我们无法确定与源关联的边的容量，因为经过每颗糖i的流无法在出边容量有1又有2的情况下作出正确的选择。

那么是否就没有办法了呢？虽然流无法在容量有1又有2的情况下作出正确的选择，但却可以在容量有1又有0的情况下最自然地作出正确的选择，流过去就表示选择了那条出边，且因为容量为1，不会再有流进入其他的出边。那么此题的构图方法也就出来了：每颗糖i作为一个点并连边(s, i, 1)，每个小孩j作为一个点并连边(j, t, floor(Bj/2))，若小孩j喜欢糖果i则连边(i, j, 1)，否则连边(i, j, 0)或者干脆不连边，效果一样。设最大流为ans，若ans+N >= ΣBj则可以满足要求。为什么？因为每颗糖迟早都要分给某个小孩，它一定会为总权值贡献1，只不过如果它分给了喜欢它的小孩就再额外贡献1。现在我只考虑这额外的1单位贡献究竟能送出去多少，最后加上基值N并与ΣBj比较即可。