D. Yukina with sticks

- First solved : 无
- 题意:n≤20个数字分成4组，求和相等，是否可行
- 数字必须用完，则最后每一组的和为$\frac{\sum a_i}{4}$，且必须除尽
- 出现单根长度> $\frac{\sum a_i}{4}$直接no
- 求出3组，剩下的木棍自然是合法的

D. Yukina with sticks

- 题意:n≤20个数字分成4组，求和相等，是否可行
- 数字必须用完，则最后每一组的和为$\frac{\sum a_i}{4}$，且必须除尽
- 出现单根长度> $\frac{\sum a_i}{4}$直接no
- 求出3组，剩下的木棍自然是合法的

- Tips:猜一猜卡时间的数据会是什么样的，对输入从大到小排序或者random_shuffle可以避开恶意数据

E. Yukina with fruits

- First solved : 无
- 题意:n种物品，每种有k_i个价值可能不同的该种物品。
- 每种物品有起购件数c_i，随便怎么买，求$\max \frac{\text{总价值}}{\max \text{单种购买个数}}$
- 显然每种物品内部按价值从大到小买
- 如果确定了购买哪几种物品与最大单种购买个数，则尽量多买
- 由于每种物品购买i个的价值是静态的，可以做个前缀和?

E. Yukina with fruits

起购件数	物品数量	价值1	价值2	价值3	价值4	价值5
1	4	5	4	3	2	\
2	3	5	4	3	\	\
3	5	5	4	3	2	1

$$\max \frac{\text{总价值}}{\max \text{单种购买个数}}$$

来算一下不同分母时的答案

单种购买数量=1，只有物品1满足起购，ans=5/1

=2，$ans=(\underline{5+4}+\underline{5+4})/2=9/1$

=3，$ans=(\underline{5+4+3}+\underline{5+4+3}+\underline{5+4+3})/3=12/1$

=4，$ans=(\underline{5+4+3+2}+\underline{5+4+3}+\underline{5+4+3+2})/4=10/1$

=5，$ans=(\underline{5+4+3+2}+\underline{5+4+3}+\underline{5+4+3+2+1})/5=41/5$

E. Yukina with fruits

起购件数	物品数量	价值1	价值2	价值3	价值4	价值5
1	4	5	4	3	2	\
2	3	5	4	3	\	\
3	5	5	4	3	2	1

$$\max \frac{\text{总价值}}{\max \text{单种购买个数}}$$

起购件数	物品数量	前1个总价值	前2	前3	前4	前5
1	4	5	9	12	14	14
2	3	0	9	12	12	12
3	5	0	0	12	14	15

来算一下不同分母时的答案

单种购买数量=1，只有物品1满足起购，ans=5/1
=2，ans=(5+4+5+4)/2=9/1
=3，ans=(5+4+3+5+4+3+5+4+3)/3=12/1
=4，ans=(5+4+3+2+5+4+3+5+4+3+2)/4=10/1
=5，ans=(5+4+3+2+5+4+3+5+4+3+2+1)/5=41/5

E. Yukina with fruits

$$\max \frac{\text{总价值}}{\max \text{单种购买个数}}$$

编号	起购件数	物品数量	前1个总价值	前2	前3	前4	前5	…	前100000
1	1	4	5	9	12	14	14		14
2	2	3	0	9	12	12	12		12
3	3	5	0	0	12	14	15		15
			…						
100000	…	100000	…						
每一列的和→			5	18	36	40	41		41

这个表维护不住了！

我们最后需要知道的只有这个表每一列的和

观察特征，物品数量少的行，后面大部分数是相同的

前缀和搞砸了！

E. Yukina with fruits

$$\max\frac{\text{总价值}}{\max\text{单种购买个数}}$$

不要了

5	18	36	40	41		41

暴力前缀和不可取！

由于最后要的只是每列的和，考虑将某种物品加入考虑，对这个求和数组的影响

求和数组sum[i]=分母为i时，能购买的最大总价值

$$\text{ans}=\max\left(\frac{sum[i]}{i}\right)$$

对于购买下限为c，物品总数为k的物品，它对求和数组的影响是？

对于所有的i∈[c,k]，sum[i]+=Σ(前i个该种物品的价值)

E. Yukina with fruits

$$\max \frac{\text{总价值}}{\max \text{单种购买个数}}$$

不要了

5	18	36	40	41		41

对于购买下限为c，物品总数为k的物品，它对求和数组的影响是?
对于所有的i∈[c,k]，sum[i]+=Σ(前i个该种物品的价值)
对于所有的i∈(k,+∞]，sum[i]+=k个该种物品的总价值

解决方案：将sum数组做成差分的形式，即新sum[i]=原sum[i]-原sum[i-1]
即如果本来的sum[i]=[1,2,4,6,9]，差分后为[1,1,2,2,3]
这样来规避在i>k后的贡献(差分后这部分对sum的贡献为0)

F. Yukina with Tree on Tree

- First solved :李扬 256(+2)
- 题意:给定一个长度为n的序列a，初始每一项均为0
  - 1 l r - 将$a_l \sim a_r$每一项加1
  - 2 l r - 依次执行第l个操作至第r个操作
- 求最后的序列
- 前置知识：还是差分
- 差分裸题：初始为0的数组，做若干次区间+1，求最后的数组
- 核心思想是只维护a[i]-a[i-1]的值，最后恢复出原数组
- 对于[l,r]整体+1，只需要a[l]++,a[r+1]--即可
- 最后恢复原数组的过程，即前缀和的过程，就是把刚刚打下的标记往后推的过程，a[l]++，那么l后面都会受影响，在r+1之后取消该影响

F. Yukina with Tree on Tree

- 题意:给定一个n长序列a，初始每一项均为0，求最后的序列
  - 1 l r - 将al ~ ar每一项加1
  - 2 l r - 依次执行第l个操作至第r个操作(保证只重复前面已有的操作)
- 显然操作1直接裸差分就可以维护，难点在操作2
- 操作2是递归的！执行第l~r个操作时可能遇到其中的2操作
- 操作1做n次，和al~ar整体+n是等价的

- 操作的顺序是无所谓的，不妨倒着做?

F. Yukina with Tree on Tree

- 题意:给定一个n长序列a，初始每一项均为0，求最后的序列
  - 1 l r - 将al ~ ar每一项加1
  - 2 l r - 依次执行第l个操作至第r个操作(保证只重复前面已有的操作)
- 操作的顺序是无所谓的，不妨倒着做?
- 因为2操作只会影响到前面的操作，维护时需要记录下每个操作被重复的次数，这又是一个区间问题。
- 还是差分，不过是从后往前的差分，记录的是a[i]-a[i+1]。
- 并且，这个差分在一边从后往前求和的过程中，一边又会修改前面的差分数组。

F. Yukina with Tree on Tree

注意ans是从前往后的差分
c是从后往前的差分

```
int n,m,op[100005],l[100005],r[100005],c[100005],ans[100005];
int main()//ans是答案序列的差分数组, c是操作应执行次数的差分数组
{
	scanf("%d%d",&n,&m);
	for(int i=1;i<=m;i++)
		scanf("%d%d%d",&op[i],&l[i],&r[i]);
	c[m+1]=1;//原c数组应是全1, 因差分只需c[m+1]=1即可
	for(int i=m;i>=1;i--)
	{
		c[i]+=c[i+1];//恢复出该操作需要做的次数
		if(op[i]==1)//1操作
			ans[l[i]]+=c[i],ans[r[i]+1]-=c[i];//给ans打标
		else
			c[r[i]]+=c[i],c[l[i]-1]-=c[i];//给c打标
	}
	for(int i=1;i<=n;i++)
		printf("%d%c",ans[i]+=ans[i-1]," \n"[i==n]);
	return 0;
}
```