Q09 落单的男女

IQ：80　目标时间：20分钟

人们聚集在某个活动会场上，根据到达会场的顺序排成一排等待入场。假设你是活动的主办人员，想把人们从队列的某个位置分成两组。

你想要让分开的两组里每一组的男女人数都均等，但如果到场顺序不对，可能出现无论怎么分，两组都不能男女均等的情况。

举个例子，有3位男性、3位女性以“男男女男女女”的顺序到场，如图7所示，无论从队列的那个位置分开，两组的男女人数都不均等。但如果到场顺序为“男男女女男女”，那么只需要在第4个人处分组就可以令分开的两组男女人数均等了。

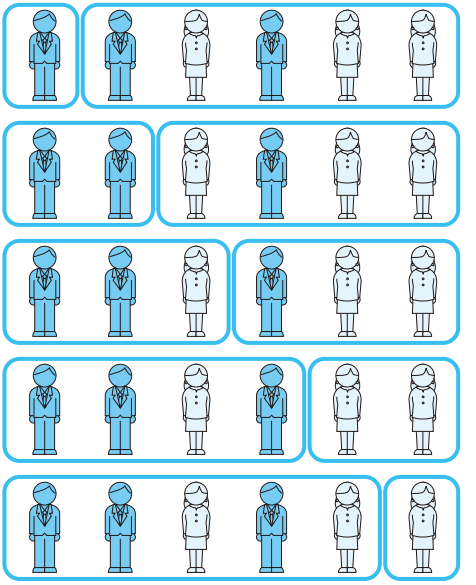

图7 男女人数无法均等的示例

问题

求男性20人、女性10人的情况下，有多少种到场顺序会导致无论怎么分组都没法实现两组男女人数均等？

把男性和女性按顺序排列，排除男女人数均等的情况。

思路

用二维表格来表示男女到场的顺序有助于我们理解。这里假设向横轴方向移动表示男性到场，向纵轴方向移动表示女性到场，那么到场顺序可以表示为一条路径。因为求的是人数不均等的情况，所以要把男女人数相等的情况先排除掉，用图表示的话则如图8所示。

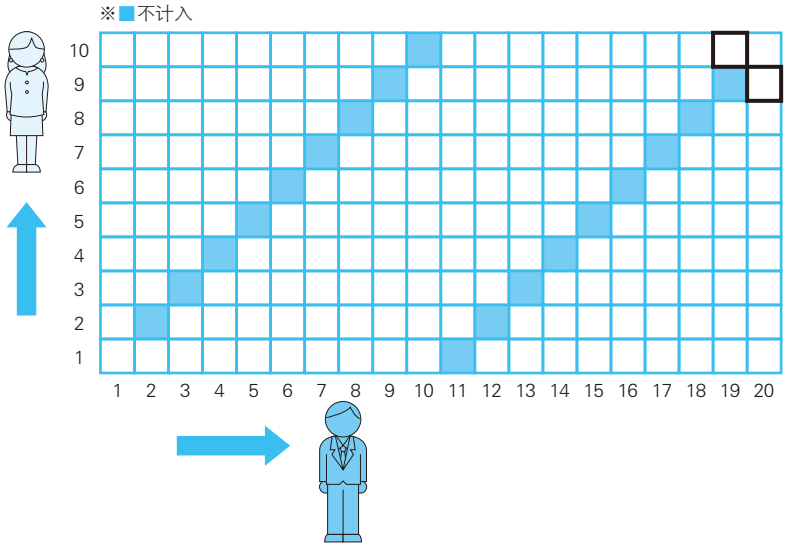

图8 把到场顺序等价于路径

只要排除图8里这两种路径即可，一种是从左下角出发会使男女人数相等的路径，另一种是从右上角出发会使男女人数相等的路径。关键就在右上角那部分吧？

Point

为使两组男女人数都不均等，只需要求到达右上角加粗的两个方格的路径，并且统计路径个数就可以了。从图表上看，也就是“求从左下角出发，到达右上角两个方格的路径有几条”。

用 Ruby 可以实现该逻辑，代码如代码清单 09.01 所示。

代码清单 09.01（q09_01.rb）

```
boy, girl = 20, 10
boy, girl = boy + 1, girl + 1
ary = Array.new(boy * girl){0}
ary[0] = 1
girl.times{|g|
  boy.times{|b|
    if (b != g) && (boy - b != girl - g) then
      ary[b + boy * g] += ary[b - 1 + boy * g] if b > 0
      ary[b + boy * g] += ary[b + boy * (g - 1)] if g > 0
    end
  }
}
puts ary[-2] + ary[-boy - 1]
```

为什么对男生和女生人数分别加1呢？

因为是从0个人开始计数的。20个人的时候，从0人到20人共21种情况；10个人的时候则是11种情况。最后统计部分可以从数组的右边数。在Ruby中，“-1”表示数组末尾。

同样的逻辑用 JavaScript 实现时，可以用多维数组描述（代码清单 09.02）。

代码清单 09.02（q09_02.js）

```
var boy = 20;
var girl = 10;
boy += 1;
girl += 1;
var ary = new Array(girl);
for (var i = 0; i < girl; i++){
  ary[i] = new Array(boy);
  for (var j = 0; j < boy; j++){
    ary[i][j] = 0;
  }
}
ary[0][0] = 1;
for (var i = 0; i < girl; i++){
  for (var j = 0; j < boy; j++){
```

```
    if ((i != j) && (boy - j != girl - i)){
      if (j > 0){
        ary[i][j] += ary[i][j - 1];
      }
      if (i > 0){
        ary[i][j] += ary[i - 1][j];
      }
    }
  }
}
console.log(ary[girl - 2][boy - 1] + ary[girl - 1][boy - 2]);
```

答案 2417416 种

Column

最短路径问题的解法

本题使用了路径的思路，这种思路常常用在最短路径问题上，也就是假设存在图 9 所示的方格，“求从 A 到 B 的最短路径有几条”的问题。

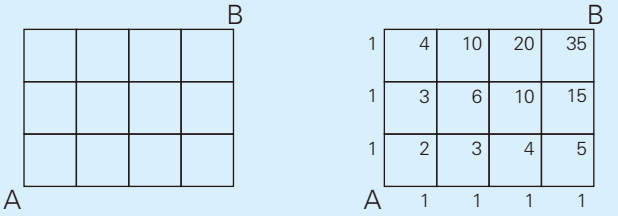

图 9 求从左下出发到达终点的路径数

横向 4 步、纵向 3 步地移动也就是“7 步中有 4 步为横向移动”，数学上就是 ${}_7C_4$，计算可得 35 种情况。本题用的是通过反复统计从最左列和最下列到达各个交叉点的情况，得到最终解的方法。对于复杂图形而言，这是一种行之有效的方法。