

## 1.休闲游戏排行榜

---

### 进阶挑战

仅使用容量为m的最大堆，每有玩家的新成绩加入，则检查是否能够插入堆，如果能插入，则更新最大堆和排行榜，否则排行不变。

## 2.魔法能量场

---

### 进阶挑战

1.增加高度后，固定双指针其中之一为该增加高度的建筑的位置，另一指针遍历其它建筑，计算各建筑组合是否有比当前最大面积更大的，此时没有左右指针之分，即没有  $(leftP < rightP)$  的限制条件。

2.算法中自动跳过了为0的情况

### 创意思考

假设游戏中除了魔法塔，还有其它能够产生增益的建筑，则玩家需要抉择当前自己更需要的资源是什么，从而决定最大化哪种建筑的摆放收益。

## 3.魔法宝箱探险

---

### 进阶挑战

1.如代码文件中第二个函数 `MaxTreasureValue_extraChance` 所示，加入另一个dp列表记录可能用过道具的情况

2. 负值已自动处理

## 4.魔法天赋评估系统

### 进阶挑战

1.假设实时更新意为，列表中会插入元素。考虑使用最大堆和最小堆分别记录属性合并后的列表的两部分。其中，最大堆存放数值低的部分，最小堆存放数值高的部分。插入数值即为比较该值与两个堆堆顶的大小，判断该插入到哪个堆中。此外需要维护两个堆的差值不超过1，且总是最大堆更大，因此当出现不满足此的情况时，需要将数量超出限制的堆的末尾/堆顶移除，转移到另一个堆中。

2.多个数组的情况需要将k个数组合并之后并找到中位数。其中，由于数组有序，可以从各数组最小值开始，选择目前的最小值，计入合并后数组。此过程只需要运行到中间部分即可（此时可以计算中位数）。由于每次仅把各数组中拿出一个元素加入比较，因此当k较大时，可以通过最小堆（容量为k）来存储当前从各个数组中抽取的元素，从而以  $O(1)$  复杂度选出当前的最小值。

### 创意思考

一个较常见的设计是根据玩家的属性，自动生成初始职业。当其通过任务或者战斗升级了属性后，只要达标便可以申请转职。同时，如果同时达到多个属性，可以职业融合。

PVP对战的匹配系统可根据玩家的属性进行匹配。