**问题描述：**

你开发了⼀个游戏 ， 日活跃用户在10万⼈以上。请设计⼀个活动排行榜系统。

在每月活动中 ，玩家得到的活动总分为 0 到 10000 之间的整数。

在每月活动结束之后 ，需要依据这⼀活动总分 ，从高到低为玩家建立排行榜。

如果多位玩家分数相同 ，则按得到指定分数顺序排序 ，先得到的玩家排在前面。

系统提供玩家名次查询接口 ，玩家能够查询自己名次前后10位玩家的分数和名次。

请使用UML图或线框图表达设计 ，关键算法可使用流程图或伪代码表达。

如果玩家分数，触发时间均相同，则根据玩家等级，名字依次排序，此情景如何设计？ （这条属于额外的答题内容，给出思路就可以）

**解题思路：**

使用redis的zset（跳表结构时间复杂度是log(n)）可以实现实现10万人的排行榜，redis的分数是一int64，由 score+时间戳共同组成，时间戳占分数的低32位，分数占剩下的高位，由于分数0-10000所以分数是 所以分数占了高14位左右。然后添加玩家分数使用zadd, 获取玩家排名前后10名的玩家可以使用zrevrange。

**优化：**

1. 如果后续需要根据玩家等级，名次等继续排名，需要在后面低位继续添加对应的分数拼接即可，
2. 这里的时间戳也是可优化的，只截取世界戳后面一些位，保证在我们项目预期时间段内有效即可。
3. 如果不是10w人的榜而是一个1000人或者100人小榜，也可以本地做一次缓存，在一定之间段内可以直接从本地缓存中拿去数据。

**代码实现：**如下nodejs版：

const Redis = require('ioredis')

const redis = new Redis()

class RankMgr {

    constructor() {

        this.redisKey = 'redisKey'

    }

    async getCombineScore(score) {

        let inow = Math.floor(Date.now() / 1000)

        score = score << 32

        return score + inow

    }

    async getPlayerRange(playerID, range = 10) {

const rank = await redis.zrevrank(this.redisKey, playerID)

        if (rank === null) {

            throw new Error('Player not found')

        }

        let start = Math.max(0, rank - range)

        let end = rank + range

        let res = await redis.zrevrange(this.redisKey, start, end, 'WITHSCORES')

        let results = []

        for (let i = 0; i < res.length; i += 2) {

            let player = res[i]

            let score = res[i + 1]

            results.push(`Player: ${player}, Score: ${score}`)

        }

        return results

    }

    async updatePlayerScore(playerID, newScore) {

        try {

            newScore = this.getCombineScore(newScore)

            await redis.zadd(this.redisKey, newScore, playerID)

        } catch (error) {

            throw new Error("update score error")

        }

    }

}