倍增&rmq

By Soda

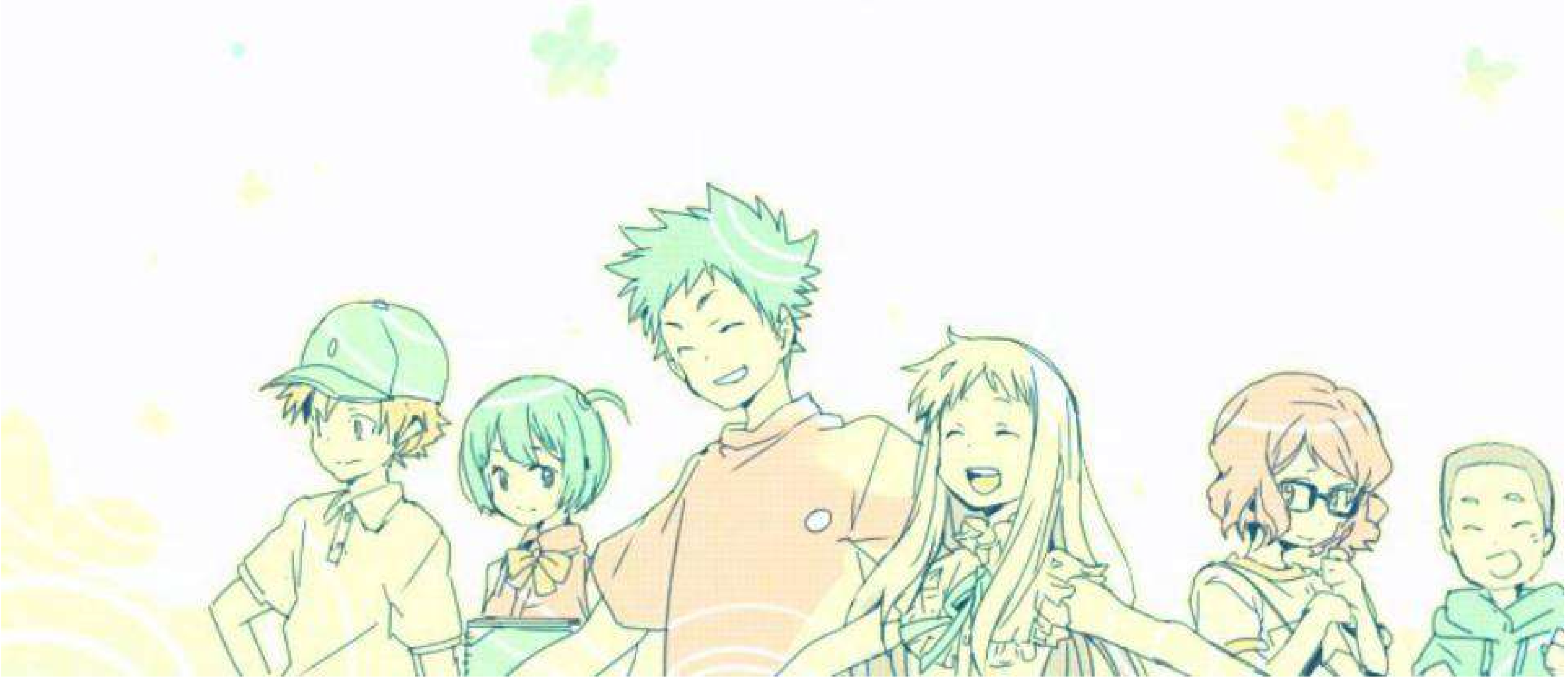
2019.7.17

倍增

- 倍增是根据已经得到的信息，将考虑的范围扩大一倍，从而加速操作的一种思想
- 使用了倍增思想的算法：
- 归并排序，快速幂
- 基于ST表的RMQ算法和树上倍增找LCA
- FFT、后缀数组等高级算法

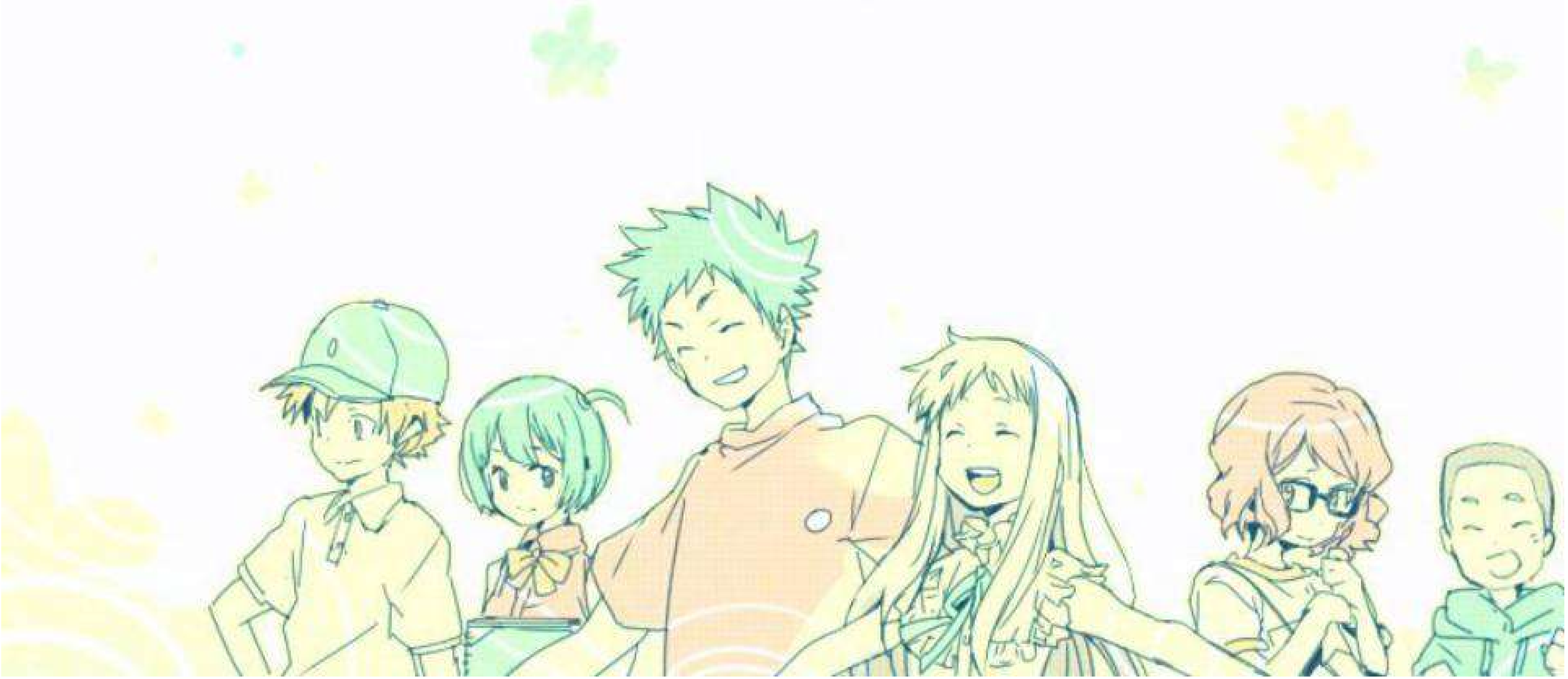
rmq

Range Maximum(Minimum) Query的缩写，顾名思义就是求某个区间的最大值或最小值，通常用在需要多次求一些区间最值的问题中。



例题1

- 输入n个数和m次询问，每次询问一个区间[L,R],求第L个数和第R个数之间的最大值。



rmq的原理

用a[1...n]表示一组数，f[i][j]表示从a[i]到a[i+2^j-1]这个范围内的最大值，也就是以a[i]为起点，连续的2^j个数中的最大值。

将2^j个数分为两部分，每一部分的元素个数就为2^(j-1)个。

整个区间的最大值，一定是左右两个区间最大值中较大的值，满足动态规划的最优原理和阶段性决策。

预处理f数组

f[i][j]=max(f[i][j-1],f[i+2^(j-1)][j-1])

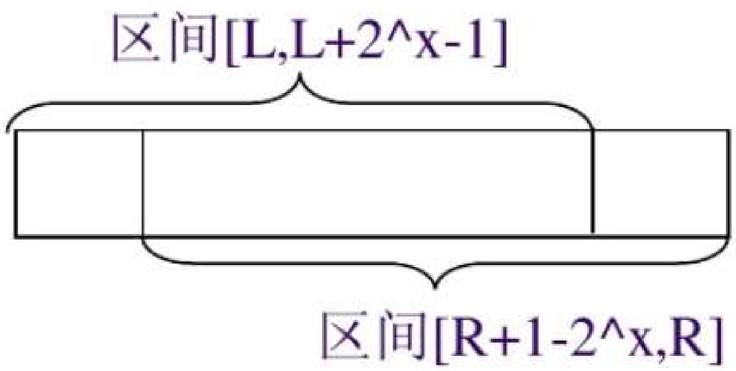
边界条件是f[i][0]=a[i]

时间复杂度O(nlgn)

查询

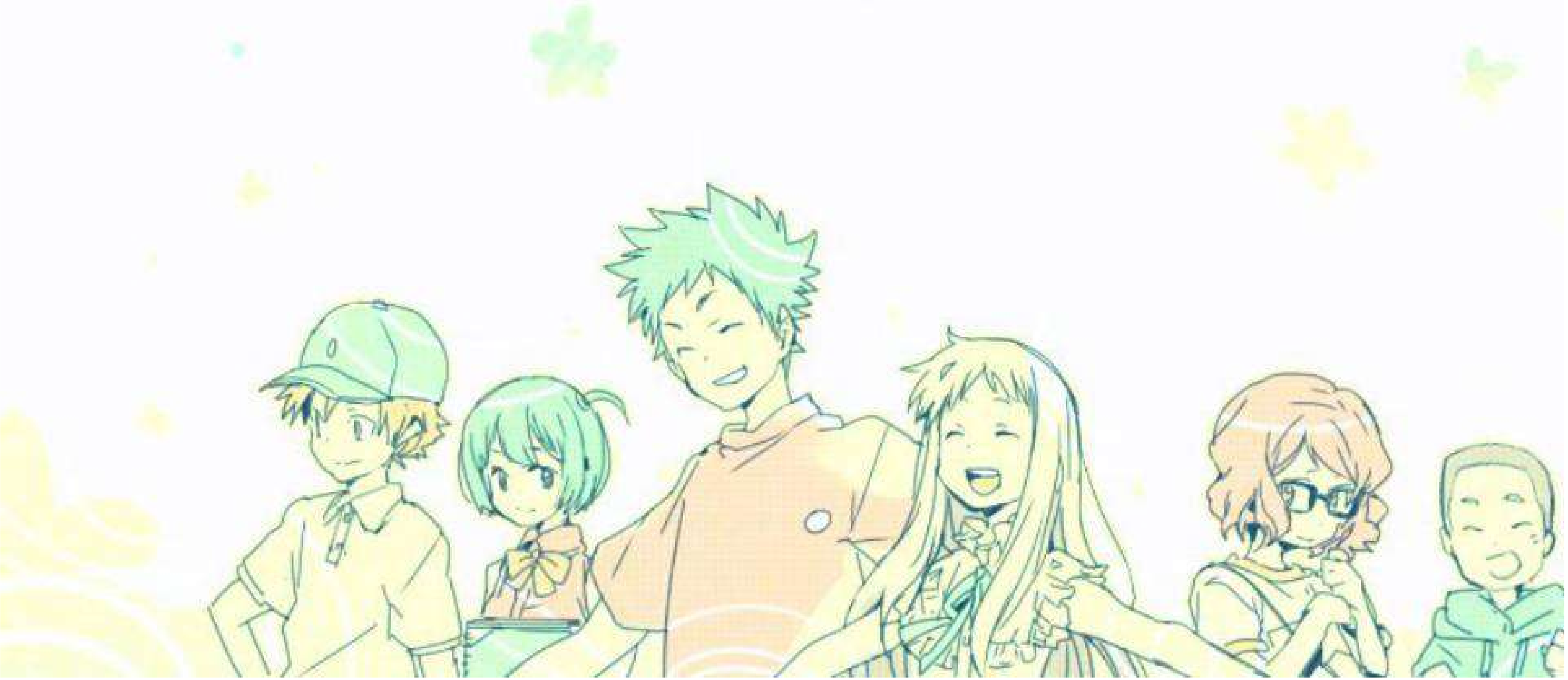
对于询问[l,r],找到最大的x,满足2^x<=r-l+1.

[l,r]=[l,l+2^x-1]U[r-2^x+1,r]



ans=max(f[l][x],f[r-2^x+1][x])

该题总的时间复杂度为O(nlgn+m)



例题2

每天,农夫 John 的N(1 <= N <= 50,000)头牛总是按同一序列排队. 有一天, John 决定让一些牛们玩一场飞盘比赛. 他准备找一群在对列中位置连续的牛来进行比赛. 但是为了避免水平悬殊,牛的身高不应该相差太大. John 准备了Q (1<=Q<=200,000)个可能的牛的选择和所有牛的身高(1<=身高<=1,000,000).他想知道每一组里面最高和最低的牛的身高差别.

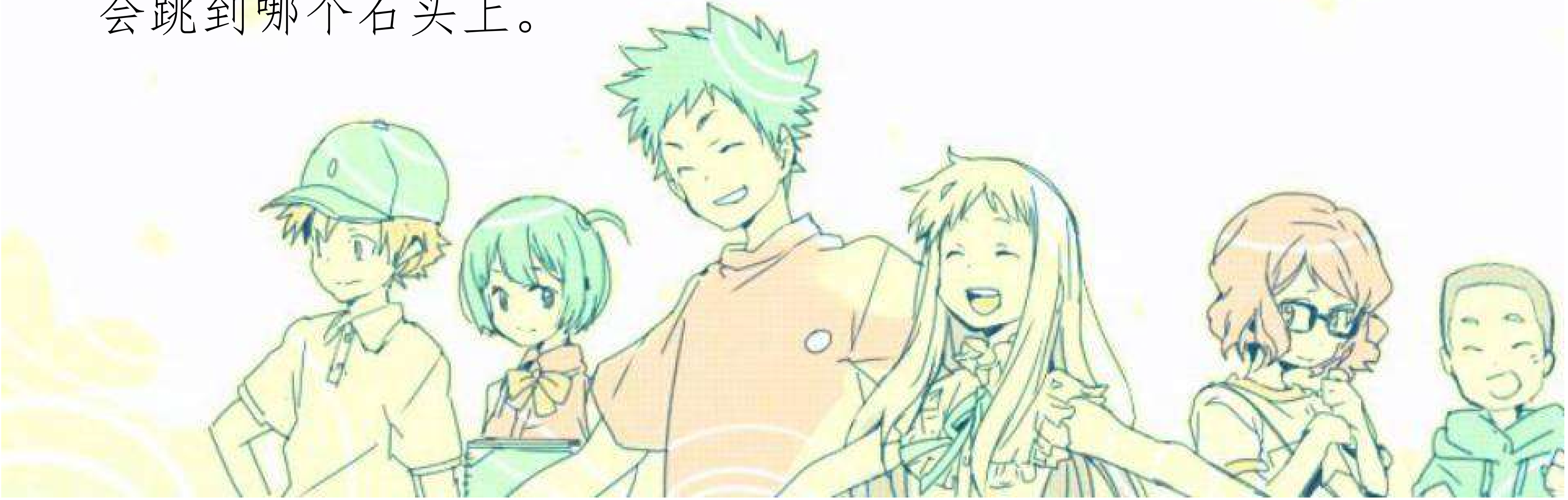
例题2

求一个数列中，一段区间内最大值和最小值的差
分别用st表维护区间的最大值和最小值
查询时分别查询并且求差即可

洛谷2880

例题3

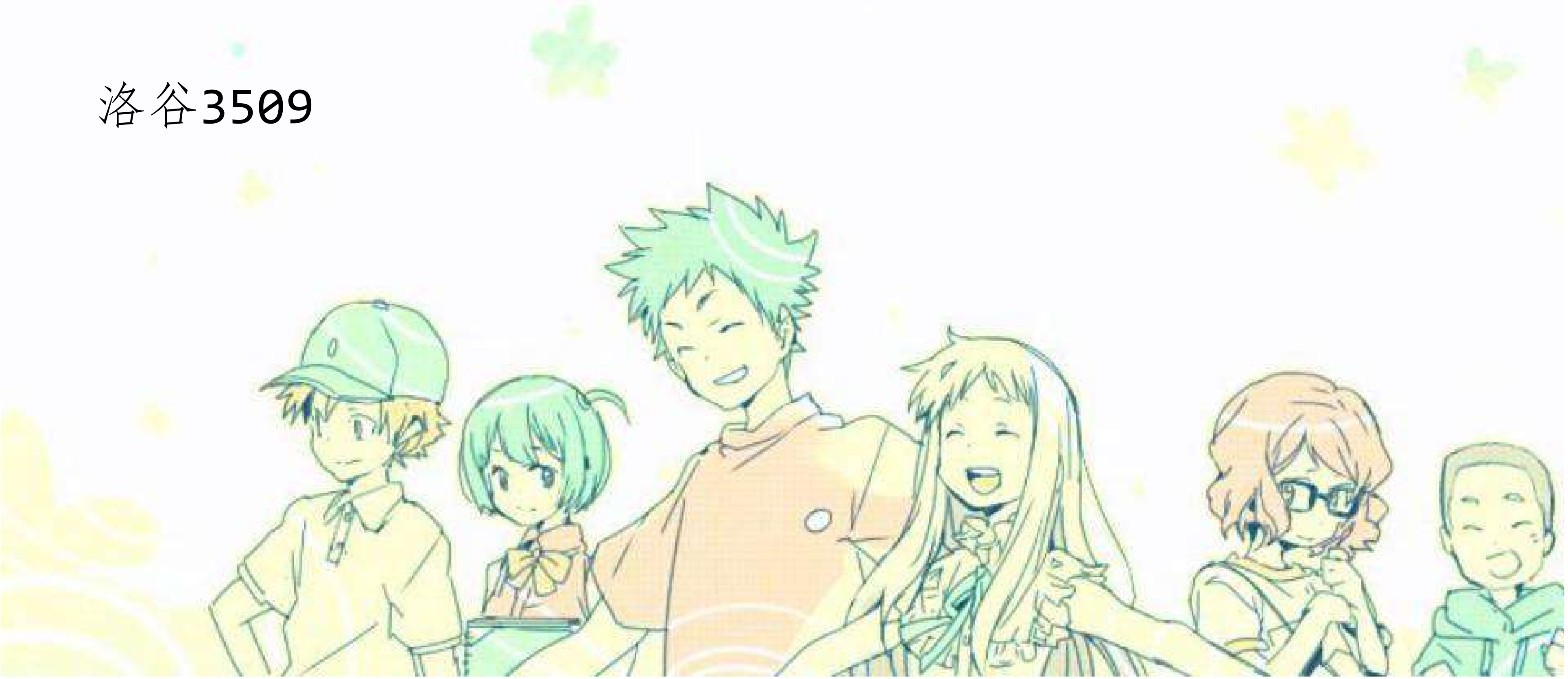
说在一条小溪上有n个石头，按照严格升序给出每个是石头到源头的距离。然后说有一群灰常SB的青蛙（每块石头上一只），每次它们会选择距离自己第k远的石头跳过去。（第i个石头和第j个石头的距离为abs(a[i] - a[j])）,如果距离当前位置第k远的石头不止一个，那就调到距离源头更近的那个石头。问跳m次之后，最开始处在i位置上的青蛙最终会跳到哪个石头上。



例题3

先求出从每个点出发跳一步到达的点，然后用倍增处理跳2^i步能到达的点

洛谷3509



例题4

小A的工作不仅繁琐，更有苛刻的规定，要求小A每天早上在6：00之前到达公司，否则这个月工资清零。可是小A偏偏又有赖床的坏毛病。于是为了保住自己的工资，小A买了一个十分牛B的空间跑路器，每秒钟可以跑2^k千米（k是任意自然数）。当然，这个机器是用longint存的，所以总跑路长度不能超过maxlongint千米。小A的家到公司的路可以看做一个有向图，小A家为点1，公司为点n，每条边长度均为一千米。小A想每天能醒地尽量晚，所以让你帮他算算，他最少需要几秒才能到公司。数据保证1到n至少有一条路径。

例题4

这道题目是最短路径与倍增算法的综合运用。

我们知道Floyed求最短路径的原理是用一个点k来修改i到j的最短距离。在这道题中，我们要灵活地用到这个方法。

因为本题中小A每秒可以跑2^k（k为任意数），所以直接求最短路径是不对的。我们可以预处理出小A一秒钟可以到达的边，这个用Floyed实现，再用一个Floyed或spfa求出1到n的最短路径就可以了。

例题4

那么关键就是如何进行预处理呢？

我们可以用一个数组F来记录，F[u][v][i]表示u到v能否通过2^i到达，这也就是1秒。在读入的时候我们就可以得出F[u][v][0]的值，然后从1~32（因为maxlongint就是2^31）枚举i，同时枚举u和v，借助Floyed用第三个点来修改的这种思想，我们再枚举一个点k，若F[u][k][i-1]和F[k][v][i-1]同时为真，则说明F[u][v][i]为真（因为2^(i-1)+2^(i-1)=2*i）。这样我们就可以与处理出所有1秒可以到的边。

然后再跑一边最短路就可以了。

洛谷1613

例题5

LYK 一开始有 n 个数，第 i 个数字是 ai,它找来了一个新的数字 P，并想将这 n 个数字中恰好一个数字替换成 P。要求替换后的最大子段和尽可能大。

注：最大子段和是指在 n 个数中选择一段区间[L,R](L<=R)使得这段区间对应的数字之和最大。

例题5

先用前缀和维护原序列

　　然后将p与序列中的每个数做差形成新的序列，用st表维护新序列的区间最大值

　　枚举序列的左右端点，然后用原序列的区间和加上新序列的区间最大值更新答案

例题6

这里有一个长度为n的正整数数列ai(下标为1~n)。并且有一个参数k。你需要找两个正整数x,y，使得x+k<=y，并且y+k-1<=n。并且要求

a[x]+a[x+1]+…+a[x+k-1]+a[y]+a[y+1]+…+a[y+k-1]最大。

对于30%的数据n<=100。

对于60%的数据n<=1000

对于100%的数据1<=n<=100000，1<=k<=n/2，1<=ai<=10^9。

例题6

题目就是让求长度为k的两个互不重叠区间的和最大

60分：处理出数列的前缀和，枚举x和y找最大和

100分：b[i]表示左端点在i长度为k的区间这个和是多少

题意即找一个x，找一个y，使得x+k<=y，并且y+k-1<=n 并且b[x]+b[y]最大

当固定x的时候,x+k<=y<=n-k+1区间求最大->线段树/倍增表 nlgn

当x从大变小的时候，y解锁了一个位置,拿这个数来更新b[y]的最大值 O(n)

例题7

她被她最近玩的一款游戏迷住了，游戏一开始有n个正整数，(2<=n<=262144)，范围在1-40。在一步中，贝西可以选相邻的两个相同的数，然后合并成一个比原来的大一的数（例如两个7合并成一个8)，目标是使得最大的数最大，请帮助Bessie来求最大值。

例题7

f[i][j] 表示从第i--f[i][j]-1 位这几位的和是 j
然后就可以递推了，因为要相同j是由前后两个相同 j-1合并而来的，

f[i][j]=f[f[i][j-1]][j-1] 合并当然是从小到大合并的

因为先把小的合并了，大的合并机会才会更多，另一个理由就是根据公式，在算 j 时 需要事先算出j-1才行

洛谷3147

例题8

有一个树状的城市网络（即n个城市由n-1条道路连接的连通图），首都为1号城市，每个城市售卖价值为ai的珠宝。

你是一个珠宝商，现在安排有q次行程，每次行程为从u城市前往v城市（走最短路径），保证v在u前往首都的最短路径上。

在每次行程开始时，你手上有价值为c的珠宝（每次行程可能不同），并且每经过一个城市时（包括u和v），假如那个城市中售卖的珠宝比你现在手上的每一种珠宝都要优秀（价值更高，即严格大于），那么你就会选择购入。

现在你想要对每一次行程，求出会进行多少次购买事件。

对于 100% 的数据，保证 $2 \le n \le 10^5, 1 \le q \le 10^5$, $1 \le a_i \le 10^5$, $1 \le c \le 10^5$。

例题8

首先可以把询问(u,v,c)转换成u点下挂个点，这个点的点权是c

现在关键问题就是如何快速找到树上每个点的father，这个father是点到根路径上最近的比它大的那个点

可以这样搞，倍增求出fa[i][j]表示i的j级父亲，mx[i][j]表示i向上到j级父亲下面这段点的最大点权

那么就可以根据mx倍增走出每个点的最近父亲，然后对于新的父亲进行倍增

对于每个询问根据深度倍增就行了

loj 6192


Thanks~