



堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

堆题目选讲

河南省实验中学信息技术组

2026年02月23日



知识回顾

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 堆
- 优先队列
- 哈夫曼树



优先队列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值
动态中位数
超市
最小序列
数据备份
合并果子
荷马史诗

练习

- 相比于普通队列先进先出 (FIFO), 优先队列 (Priority queue) 每个元素都有一个优先级, 按照优先级高低确定出队顺序的, 优先级高的先出队。
- 优先队列可以用堆来实现, 堆顶为优先级最大的元素。
- STL 中提供了一个优先队列的实现 `priority_queue`, 使用时需要引用头文件 `#include <queue>`。

操作	功能
<code>priority_queue<int> pq;</code>	建立一个存放 <code>int</code> 类型数据的优先队列
<code>pq.empty()</code>	判断优先队列是否为空
<code>pq.size()</code>	返回优先队列大小
<code>pq.top()</code>	返回队头元素
<code>pq.push(int x)</code>	插入元素
<code>pq.pop()</code>	删除队头元素

表: 优先队列操作说明



优先队列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值
动态中位数
超市
最小序列
数据备份
合并果子
荷马史诗

练习

```
1 int a[9] = { 0, 1, 2, 3, 4, 8, 9, 3, 5 }; // 定义数组
2 // 定义优先队列, 并将数组 a[1:8] 的值复制进去 默认建立最大堆
3 priority_queue<int> pq(a + 1, a + 9);
4 cout << pq.size(); // 输出 8, 优先队列中有 8 个元素
5 cout << pq.top(); // 输出 9, 队列中优先级最高的为 9
6 //输出队列的前 6 个元素, 输出 9 8 5 4 3 3, 从大到小
7 for (int i = 1; i <= 6; i++)
8 {
9     cout << pq.top() << " ";
10    pq.pop();
11 }
12 //此时队列中只剩下 2 1
13 pq.push(7); // 新元素 7 入队
14 cout << pq.size(); // 输出 3, 优先队列中有 3 个元素
15 cout << pq.top(); // 输出队列头元素为 7, 因为队列中 7 最大
16
17 //清空队列, 并依次输出队中元素, 从大到小输出 7 2 1
18 while (!pq.empty())
19 {
20     cout << pq.top() << " ";
21     pq.pop();
22 }
```



优先队列

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数
动态中位数
超市
最小序列
数据备份
合并果子
荷马史诗

练习

- 如果能让队列按照从小到大的顺序排队呢？
- 方法 1: 把数据的值全部变成相反数，然后放在队列中，那么就可以从小到大排序，但是要注意出队入队都要变成相反数。
- 方法 2: 定义从小到大排序的队列。例如建立一个全为整数最小堆，写为 `priority_queue<int, vector<int>, greater<int> > pq;`¹。

¹`greater<int>`和最后一个`>`之间要有空格，是为了与输出符号`>>`区分，在 C++11 及其以后的标准中可以省去空格。



优先队列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 如果元素值有多个关键字怎么办?
- 如果只有两个关键字, 那么可以考虑使用 `pair`, 那么将按照 `pair` 的 `first` 元素从大到小排队, 如果 `first` 相同将按照 `second` 元素从大到小排队。
- 当然如果你想从小到大排序, 可以将值改为相反数, 也可以定义从小到大排队的队列。

```
1 pair<int, int> a = {3, 5}; // a.first=3 是第一个元素 a.second=5 是第二个元素
2 priority_queue<pair<int,int> > pq; // 按照 first 和 second 从大到小排序
3 pq.push({1, 2});
4 pq.push(a);
5 pair<int, int> b = pq.top();
6
7 // 按照 first 和 second 从小到大排序
8 priority_queue<pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>> pq;
```



优先队列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 如果有三个关键字以上，怎么办？²
- 自己实现堆。
- 使用多个 pair 嵌套。

```
1 priority_queue<pair<pair<int,int>, int>> pq; // 3 个关键字  
2 priority_queue<pair<pair<int,int>, pair<int,int>>> pq; // 4 个关键字
```

- 定义结构体并重载小于比较方式。

```
1 struct Node  
2 {  
3     int x, y, z;  
4     bool operator<(const Node &b) const  
5     {  
6         return a.x > b.x || a.x == b.x && a.y > b.y;  
7     }  
8 };  
9  
10 priority_queue<Node> pq; // 按照 x, y 从大到小排队
```

²自行学习



【例】最小函数值

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【题目描述】

有 n 个函数，分别为 f_1, f_2, \dots, f_n ，定义 $f_i(x) = a_i x^2 + b_i x + c_i (x \geq 1)$ 。给定这些 a_i, b_i, c_i ，请求出所有函数的所有函数值中最小的 m 个 (如有重复的要输出多个)。

【输入格式】

第一行输入两个正整数 $n, m (n, m \leq 5 \times 10^5)$ 。

以下 n 行每行三个正整数，其中第 i 行的三个数分别为 $a_i, b_i, c_i (a_i, b_i, c_i \geq 0)$ 。

【输出格式】

输出将这 n 个函数所有可以生成的函数值排序后的前 m 个元素。

这 m 个数应该输出到一行，用空格隔开，并且最后一个数右侧也有一个空格。



【例】最小函数值

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【样例输入】

```
3 10
4 5 3
3 4 5
1 7 1
```

【样例输出】

```
9 12 12 19 25 29 31 44 45 54
```

【样例解释】

对于函数 $f_1(x) = 4x^2 + 5x + 3$ ，它的最小 10 个函数值为
12, 29, 54, 87, 128, 177, 234, 299, 372, 453;

对于函数 $f_2(x) = 3x^2 + 4x + 5$ ，它的最小 10 个函数值为
12, 25, 44, 69, 100, 137, 180, 229, 284, 345;

对于函数 $f_3(x) = x^2 + 7x + 1$ ，它的最小 10 个函数值为
9, 19, 31, 45, 61, 79, 99, 121, 145, 171;

那么最小的前 10 个函数值是 9, 12, 12, 19, 25, 29, 31, 44, 45, 54。



【例】最小函数值

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 每个函数的函数值是递增的，但是彼此之间的函数值大小则无确定关系。
- 而只求最小的 m 个函数值，那么每个函数最多只要求前 m 个函数值。
- 将求出的 $n \times m$ 个函数值从小到大排序，取前 m 个即可。
- 时间复杂度： $O(MN \log(MN))$ ，空间复杂度： $O(MN)$ ，超时并且内存超限。



【例】最小函数值

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 首先，考虑最小的函数值，必定是所有函数 $x = 1$ 时的函数值的最小值，设这个函数为 f_k 。
- 其次，第二小的函数值必定除了 f_k 之外的所有函数 $x = 1$ 时的函数值和函数 f_k 当 $x = 2$ 时的函数值的最小值，其他函数值依次类推。
- 那么每次只需要保留每个函数的一个函数值 (最小的) 即可，每次在这些函数值中找最小的就是答案。
- 可以将函数值放在最小堆中，那么堆顶即为最小函数值，当堆顶的最小值被输出后，就继续计算该函数的下一个函数值放入堆中即可。
- 但是只放函数值是不够的，因为每一次还要知道最小的函数值是哪个函数，所以堆内需要同时存储函数值和函数编号。



【例】最小函数值

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 int x[N];          // 每个函数当前的自变量值
2 long long f(int a, int b, int c, int x)
3 {
4     return (long long)a * x * x + b * x + c;
5 }
6 // 堆内需要存储函数值和函数编号
7 priority_queue<pair<long long, int>, vector<pair<long long, int> >,
8               greater<pair<long long, int> > > pq;
9 for(int i = 1; i <= n; ++i) // 先求每个函数 x=1 时的值
10 {
11     pq.push({f(a[i], b[i], c[i], 1), i});
12     x[i] = 1;
13 }
14 for(int i = 1; i <= m; ++i)
15 {
16     pair<long long, int> t = pq.top(); pq.pop(); // 取堆顶
17     cout << t.first << " ";
18     int k = t.second; // 当前函数编号
19     pq.push({f(a[k], b[k], c[k], x[k] + 1), k}); // 计算下一个函数值
20     x[k] = x[k] + 1;
21 }
```

- 时间复杂度: $O(M \log N)$ 。



【例】动态中位数

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【题目描述】

读入一个整数序列，每当已经读入的整数个数为奇数时，输出已读入的整数构成的序列的中位数。

【输入格式】

第 1 行一个整数 n ($n \leq 50000$) 且 n 一定为奇数。

第 2 行 n 个整数，表示读入的序列。

【输出格式】

一行若干个整数，表示依次输出的中位数。

【样例 1 输入】

```
9
1 2 3 4 5 6 7 8 9
```

【样例 2 输入】

```
23
23 41 13 22 -3 24 -31 -11 -8 -7
3 5 103 211 -311 -45 -67 -73 -81 -99
-33 24 56
```

【样例 1 输出】

```
1 2 3 4 5
```

【样例 1 输出】

```
23 23 22 22 13 3 5 5 3 -3 -7 -3
```



【例】动态中位数

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 定义最大堆维护较小的值，定义最小堆维护较大的值，那么两个堆堆顶值即为中间值。
- 对于序列中第 i 个数 a_i ，如果它比最小堆的堆顶大，那么插入最小堆，否则插入最大堆。
- 要求最小堆的大小要等于最大堆的大小或者等于最大堆的大小 + 1。
- 那么当序列中的数输入奇数个时，最小堆的堆顶即为中位数。
- 时间复杂度： $O(N \log N)$ 。



【例】动态中位数

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 priority_queue<int> maxp; // 最大堆
2 priority_queue<int, vector<int>, greater<int> > minp; // 最小堆
3 for(int i = 1; i <= n; ++i)
4 {
5     if(minp.empty() || minp.top() < a[i]) minp.push(a[i]);
6     else maxp.push(a[i]);
7     while(maxp.size() > minp.size())
8         minp.push(maxp.top()), maxp.pop()
9     while(minp.size() - maxp.size() > 1)
10        maxp.push(minp.top()), minp.pop();
11    if(i % 2) cout << minp.top() << " ";
12 }
```



【例】超市

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【题目描述】

超市里有 n 件商品，每个商品都有利润 p_i 和过期时间 d_i ，每天只能卖一件商品，过期商品 (在第 k 天， $k > d_i$) 不能再卖。

求合理安排每天卖的商品的情况下，可以得到的最大收益是多少。

【输入格式】

第 1 行一个整数 n ($n \leq 10000$)。

接下来 n 行，每行两个整数表示商品的利润和过期时间 ($1 \leq p_i, d_i \leq 10000$)。

【输出格式】

一行一个整数表示最大收益值。

【样例输入】

```
7
20 1
2 1
10 3
100 2
8 2
5 20
50 10
```

【样例输出】

```
185
```



【例】超市

堆

河南省实验中学
信息技术组

优先队列

例题

最小两数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 商品过期前能卖尽量卖，但是由于每天只能卖一件，而商品又特别多，所以第 t 天时，可能会有很多商品过期不能卖掉。所以，第 t 天时，在保证不卖出过期商品的前提下，尽量卖出利润前 t 大的商品。
- 把商品按照过期时间从小到大排序，然后依次处理每个商品。
- 为了始终维护利润前若干大的商品，建立一个最小堆来存储售卖的商品 (结点为利润)。
- 当处理到第 i 件商品时：
 - 如果 d_i 大于当前堆大小，那么直接将该商品的利润入堆 (过期前肯定有时间可以卖)。
 - 如果 d_i 等于当前堆大小，说明在目前方案中，前 d_i 天已经安排了 d_i 个商品卖出。此时，如果当前商品利润大于堆顶值 (已经安排的 d_i 个商品中的最低利润)，那么用当前商品的利润替换堆顶 (用当前商品替换掉原方案中利润最低的商品)。
- 堆内存储的即为要卖出商品的利润，堆中所有元素的和就是答案。



【例】超市

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 sort(g + 1, g + n + 1, cmp); // 按照过期时间从小到大排序
2 priority_queue<int, vector<int>, greater<int> > pq;
3 for(int i = 1; i <= n; ++i)
4 {
5     // 前 d 天中所有合法商品的前 d 大未找完
6     if(pq.size() < g[i].d) pq.push(g[i].p);
7     // 前 d 大已经确定 但是第 i 件物品更优
8     else if(pq.size() == g[i].d && pq.top() < g[i].p)
9     {
10         pq.pop();
11         pq.push(g[i].p);
12     }
13 }
14 int ans = 0;
15 while(!pq.empty()) ans += pq.top(), pq.pop();
```



【例】最小序列

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【题目描述】

给定 m 个序列，每个包含 n 个非负整数。

现在我们可以从每个序列中选择一个数字以形成具有 m 个整数的序列。

很明显，我们一共可以得到 n^m 个这种序列，然后我们可以计算每个序列中的数字之和，并得到 n^m 个值。

现在请你求出这些序列和之中最小的 n 个值。

【输入格式】

第一行输入两个整数 $m(1 \leq m \leq 100)$ 和 $n(1 \leq n \leq 2000)$ 。

接下在 m 行输入 m 个整数序列，数列中的整数均不超过 10000。

【输出格式】

递增顺序输出最小的 n 个序列和，数值之间用空格隔开。

【样例输入】

```
2 3
1 2 3
2 2 3
```

【样例输出】

```
3 3 4
```



【例】最小序列

堆

河南省实验中学
信息技术组

优先队列

例题

最小两数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 首先考虑两个序列的问题，即从两个序列中任取一个数相加构成的 n^2 个和中选出前 n 小的和。

- 设这两个序列为 a, b ，首先将它们分别从小到大排序。

- 最小的和一定是 $a_1 + b_1$ ，次小和为 $\min(a_1 + b_2, a_2 + b_1)$ 。

- 假设次小和为 $a_1 + b_2$ ，那么第 3 小和为 $\min(a_2 + b_1, a_2 + b_2, a_1 + b_3)$ ；

- 假设次小和为 $a_2 + b_1$ ，那么第 3 小和为 $\min(a_1 + b_2, a_2 + b_2, a_3 + b_1)$ 。

也就是说当确定 $a_i + b_j$ 为第 k 小后， $a_{i+1} + b_j$ 和 $a_i + b_{j+1}$ 就加入了第 $k + 1$ 小和的备选答案集合。

- 需要注意的是， $a_1 + b_2$ 和 $a_2 + b_1$ 都能产生 $a_2 + b_2$ 这个备选答案。于是为了避免重复，规定：如果是从 j 加 1 产生的备选答案，以后只能增加 j ，不能再增加 i ³。

³从 $a_1 + b_1$ 到 $a_i + b_j$ 必须先增加 i 再增加 j ，保证了备选答案产生序列的唯一性。



【例】最小序列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值
动态中位数
超市
最小序列
数据备份
合并果子
荷马史诗

练习

- 建立一个最小堆，结点存储三元组 $(a_i + b_j, i, j, last)$ ，其中 $last$ 表示上次增加的是否是 j 。
- 首先，堆中只有 $(a_1 + b_1, 1, 1, false)$ 。
- 取出堆顶 $(a_i + b_j, i, j, last)$ ，堆顶的 $a_i + b_j$ 即为要求的结果，然后把 $(a_i + b_{j+1}, i, j + 1, true)$ 入堆，如果 $last$ 为 $false$ ，再把 $(a_{i+1} + b_j, i + 1, j, false)$ 入堆。
- 重复上一步 n 次，每次取出的堆顶的权值构成前 n 小和。
- 于是对于 m 个序列，首先将 1 号和 2 号序列执行上述算法求出前 n 小的序列，然后将 3, 4, ..., m 号序列依次与上述结果序列执行算法后构成前 n 小的序列即可。
- 时间复杂度： $O(MN \log N)$ 。



【例】最小序列

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 // 将 a 和 b 合并到 c
2 sort(a + 1, a + n + 1);
3 sort(b + 1, b + n + 1);
4 memset(c, 0, sizeof(c));
5 priority_queue<Node> pq;
6 pq.push({a[1] + b[1], 1, 1, false});
7 for(int k = 1; k <= n; ++k)
8 {
9     auto t = pq.top(); pq.pop();
10    c[k] = t.x;
11    int i = t.i, j = t.j;
12    // true 表示移动的是指针 j
13    pq.push({ a[i] + b[j + 1], i, j + 1, true });
14    // false 表示移动的是指针 i
15    if(!t.last) pq.push({ a[i + 1] + b[j], i + 1, j, false });
16 }
```



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

你在一家 IT 公司为大型写字楼或办公楼的计算机数据做备份。然而数据备份的工作是枯燥乏味的，因此你想设计一个系统让不同的办公楼彼此之间互相备份，而你则坐在家中尽享计算机游戏的乐趣。

已知办公楼都位于同一条街上。你决定给这些办公楼配对（两个一组）。每一对办公楼可以通过在这两个建筑物之间铺设网络电缆使得它们可以互相备份。

然而，网络电缆的费用很高。当地电信公司仅能为你提供 k 条网络电缆，这意味着你仅能为 k 对办公楼（或总计 $2k$ 个办公楼）安排备份。任一个办公楼都属于唯一的配对组（换句话说，这 $2k$ 个办公楼一定是相异的）。

此外，电信公司需按网络电缆的长度（公里数）收费。因而，你需要选择这 ks 对办公楼使得电缆的总长度尽可能短。换句话说，你需要选择这 k 对办公楼，使得每一对办公楼之间的距离之和（总距离）尽可能小。



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【输入格式】

第一行包含整数 n 和 k ，其中 $n(1 \leq n \leq 10^5)$ 表示办公楼的数目， $k(1 \leq k \leq \frac{n}{2})$ 表示可利用的网络电缆的数目。

接下俩一行 n 个整数，表示办公楼到大街起点处的距离，这些整数将按照从小到大的顺序依次出现，保证距离不超过 10^9 。

【输出格式】

一行只包含一个整数，也就是网络电缆的最小总长度。

【样例输入】

```
5 2
1 3 4 6 12
```

【样例输出】

```
4
```



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

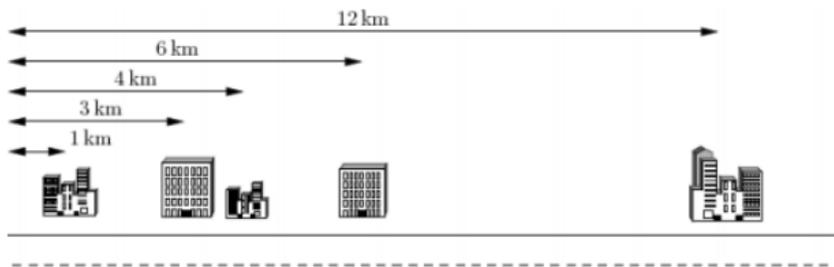
合并果子

荷马史诗

练习

【样例解释】

这 5 个办公楼分别位于距离大街起点 1km, 3km, 4km, 6km 和 12km 处。电信公司仅为你提供 $k = 2$ 条电缆。



上例中最好的配对方案是将第 1 个和第 2 个办公楼相连，第 3 个和第 4 个办公楼相连。这样可按要求使用 $k = 2$ 条电缆。第 1 条电缆的长度是 $3km - 1km = 2km$ ，第 2 条电缆的长度是 $6km - 4km = 2km$ 。这种配对方案需要总长 $4km$ 的网络电缆，满足距离之和最小的要求。



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小两数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 显然，两个配对的办公楼一定是相邻的。
- 设这 n 个办公楼之间的距离分别为 d_1, d_2, \dots, d_{n-1} ，那么问题转化为：在 $n-1$ 个数中选 k 个数使得它们的和最小，且这 k 个数不相邻。
- 将这 k 个数放在最小堆中，每次选最小值，判定和以前选择的数是否相邻 (错误贪心)。



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 当 $k = 1$ 时，显然选最小值即可。
- 当 $k = 2$ 时，答案一定是如下两种情况之一⁴：
 - ① 选择最小值 d_i 以及除 d_{i-1}, d_i, d_{i+1} 之外的最小值；
 - ② 选择最小值 d_i 左右两侧的 d_{i-1}, d_{i+1} 。
- 那么在最优解中，最小值左右两侧的数要么同时选，要么都不选。
- 可以先选上最小值 d_i ，然后删除 d_{i-1}, d_i, d_{i+1} ，然后再将 $d_{i-1} + d_{i+1} - d_i$ 插入到原来 d_i 的位置，然后求解剩余数中选择 $k - 1$ 个数的问题⁵。
- 为了方便删除和插入数值，可以用链表维护序列的值；同时建立最小堆维护当前最小的值及其位置。
- 每次只需要取出堆顶，将堆顶累加到答案中，设堆顶的链表下标为 p ，然后删除 p 的左右两个结点（堆内也标记为删除），修改结点 p 的值为左右两个结点值之和减去 p 的值，并将该值入堆。

⁴证明：如果 d_{i-1} 和 d_{i+1} 都没有选，那么不选最小值 d_i 一定不优；如果 d_{i-1} 和 d_{i+1} 只选了一个，那么把选了的那个换成 d_i ，答案也会变小。

⁵反悔贪心：如果后续选择了 $d_{i-1} + d_{i+1} - d_i$ ，相当于抛弃 d_i ，选择 d_{i-1}, d_{i+1} 。



【例】数据备份

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 while(cnt < k)
2 {
3     auto t = pq.top(); pq.pop();
4     int k = t.second, l = a[k].l, r = a[k].r;
5     if(!in[k]) continue;
6     ans += t.first;
7     // 如果是端点 那么端点被选 端点的相邻位置一定不会被选
8     if(!l) del(r), del(k); // 如果是左端点
9     else if(!r) del(l), del(k); // 如果是右端点
10    else
11    {
12        // 用  $D[l]+D[r]-D[k]$  代替  $D[k]$ 
13        a[k].x = a[l].x + a[r].x - a[k].x;
14        pq.push({ a[k].x, k });
15        del(l), del(r);
16    }
17    ++cnt;
18 }
```



【例】合并果子

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

在一个果园里，多多已经将所有果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 $n - 1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1, 2, 9。可以先将 1, 2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力为 $3 + 12 = 15$ 。可以证明 15 为最小的体力耗费值。



【例】合并果子

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数位

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【输入格式】

第一行一个整数 n ($n \leq 10000$), 表示果子的种类数。

第二行包含 n 个整数, 用空格分隔, 第 i 个整数 a_i ($1 \leq a_i \leq 20000$) 是第 i 种果子的数目。

【输出格式】

一行只包含一个整数, 也就是最小的体力耗费值。

【样例输入】

```
3
1 2 9
```

【样例输出】

```
15
```



【例】合并果子

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

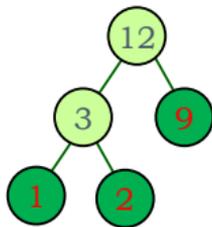
数据备份

合并果子

荷马史诗

练习

- 每次合并消耗的体力等于两堆果子的重量之和，所以最终消耗的体力总和就是每堆果子的重量乘上它参与合并的次数。
- 这恰好是一个哈夫曼树，果子的重量就是叶子结点的权值，参与合并的次数就是叶子结点到根结点的距离。



- 利用果子的重量建立一个最小堆，不断取出堆中最小的两个值，将这两个值累加到答案(消耗的体力)，同时将它们的和插入堆中。
- 当堆的大小为 1 时，算法结束，输出累加的答案即可。



【例】荷马史诗

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

一部《荷马史诗》中有 n 种不同的单词，从 1 到 n 进行编号。其中第 i 种单词出现的总次数为 w_i 。Allison 想要用 k 进制串 s_i 来替换第 i 种单词，使得其满足如下要求：

对于任意的 $1 \leq i, j \leq n$ ， $i \neq j$ ，都有： s_i 不是 s_j 的前缀。

现在 Allison 想要知道，如何选择 s_i ，才能使替换以后得到的新的《荷马史诗》长度最小。在确保总长度最小的情况下，Allison 还想知道最长的 s_i 的最短长度是多少？

一个字符串被称为 k 进制字符串，当且仅当它的每个字符是 0 到 $k-1$ 之间（包括 0 和 $k-1$ ）的整数。

字符串 $str1$ 被称为字符串 $str2$ 的前缀，当且仅当：存在 $1 \leq t \leq m$ ，使得 $str1 = str2[1..t]$ 。其中， m 是字符串 $str2$ 的长度， $str2[1..t]$ 表示 $str2$ 的前 t 个字符组成的字符串。



【例】荷马史诗

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

【输入格式】

第 1 行包含 2 个正整数 n, k ($n \leq 10^5, k \leq 9$)，中间用单个空格隔开，表示共有 n 种单词，需要使用 k 进制字符串进行替换。

第 2 行 n 个整数，第 i 个数 w_i ($w_i \leq 10^{11}$) 表示第 i 个单词的出现次数。

【输出格式】

第 1 行输出 1 个整数，为《荷马史诗》经过重新编码以后的最短长度。

第 2 行输出 1 个整数，为保证最短总长度的情况下，最长字符串 s_i 的最短长度。

【样例 1 输入】

```
4 2
1 1 2 2
```

【样例 2 输入】

```
6 3
1 1 3 3 9 9
```

【样例 1 输出】

```
12
2
```

【样例 2 输出】

```
36
3
```



荷马史诗

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数位

动态中位数

超市

最小序列

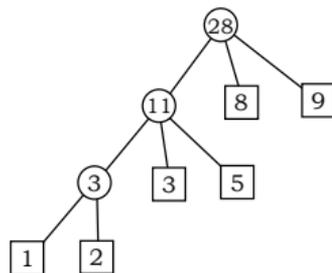
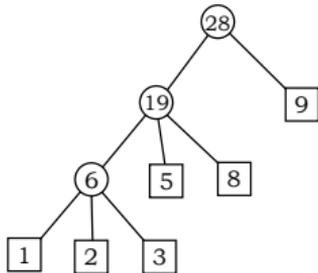
数据备份

合并果子

荷马史诗

练习

- 本题所构造的编码方式就是哈夫曼编码，将单词出现的次数 $w_1 \sim w_n$ 作为哈夫曼树的叶子结点，建立一棵 k 叉哈夫曼树，每个结点的 k 个分支分别标记字符 $0 \sim k-1$ 即可。
- 对于 k 叉哈夫曼树，最直观的方法是每次从堆中取出最小的 k 个权值构造树，这种方法可行吗？
- 仔细思考会发现，如果在最后一次构造树时，堆的大小 (堆中树的个数) 在 $2 \sim k-1$ 之间，那么整个哈夫曼树的根的子结点数就少于 k ，这显然不是最优解，因为此时如果任取一个深度稍大的结点，将它改为根结点的子结点，一定会更优。
- 例如，左图的带权路径之和为 53，右图的带权路径之和为 42。





荷马史诗

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

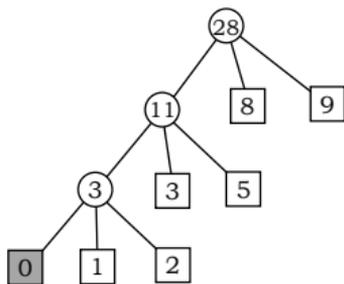
数据备份

合并果子

荷马史诗

练习

- 在执行建树之前，可以补加一些权值为 0 的叶子结点，使得叶子结点的个数满足 $(n - 1) \bmod (k - 1) = 0^6$ ，然后执行“每次从堆中取最小的 k 个权值”的操作建树就是正确的。



- 本题还要求最长的编码 s_i 长度最短，那么在构造哈夫曼树时，对于权值相同的结点，优先考虑深度小的结点合并即可。

⁶让所有的内部结点都有 k 个孩子，设叶子结点有 n 个，内部结点有 m 个，则 $km = m + n - 1 \Rightarrow n - 1 = m(k - 1) \Rightarrow (n - 1) \bmod (k - 1) = 0$ 。



【例】荷马史诗

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

```
1 struct Node
2 {
3     long long w; // 权值
4     int d; // 树深度
5     bool operator<(const Node &b) const
6     {
7         return w > b.w || w == b.w && d > b.d;
8     }
9 };
10 priority_queue<Node> pq(a + 1, a + n + 1);
11 while((n - 1) % (k - 1)) pq.push({0, 0}), ++n; // 构造空结点
12 long long ans = 0;
13 while(pq.size() > 1)
14 {
15     Node r = {0, 0};
16     for(int i = 1; !pq.empty() && i <= k; ++i)
17     {
18         Node t = pq.top(); pq.pop();
19         r.w += t.w;
20         r.d = max(r.d, t.d + 1);
21     }
22     ans += r.w;
23     pq.push(r);
24 }
```



练习

堆

河南省实验中学
信息技术组

优先队列

例题

最小函数值

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 最小函数值 (COGS 2334)
- Stall Reservation(COGS 3448)
- Supermarket(COGS 3097)
- Runing Median(COGS 3415)
- 黑盒子 (COGS 601)
- 数字对数 (COGS 3759)
- 鱼塘钓鱼 (COGS 1748)
- 廊桥分配 [CSP-S 2021](COGS 3619)
- 建筑抢修 [JSOI 2007](COGS 929)
- 舞蹈课 (COGS 3296)



练习

堆

河南省实验中学
信息技术组

优先队列

例题

最小两位数

动态中位数

超市

最小序列

数据备份

合并果子

荷马史诗

练习

- 序列合并 (洛谷 P1631)
- Sequence(COGS 3461)
- 数据备份 [CTSC 2007](COGS 2884)
- 种树 [国家集训队 2011](COGS 1862)
- 生日礼物 (COGS 3477)
- 合并果子 [NOIP 2004](COGS 75)
- 哈夫曼编码的长度 (COGS 3192)
- 哈夫曼编码 (COGS 3298)
- 荷马史诗 [NOI 2015](COGS 2021)