

全国青少年信息学奥林匹克竞赛

CCF NOI 2023

第二试

时间：2023 年 7 月 26 日 08:00 ~ 13:00

题目名称	贸易	字符串	合并书本
题目类型	传统型	传统型	传统型
目录	trade	string	book
可执行文件名	trade	string	book
输入文件名	trade.in	string.in	book.in
输出文件名	trade.out	string.out	book.out
每个测试点时限	1.5 秒	1.0 秒	1.5 秒
内存限制	512 MiB	512 MiB	512 MiB
测试点数目	20	25	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	trade.cpp	string.cpp	book.cpp
-----------	-----------	------------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 因违反以上两点而出现的错误或问题，申诉时一律不予受理。
4. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
5. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 只提供 Linux 格式附加样例文件。
8. 禁止在源代码中改变编译器参数（如使用 #pragma 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
9. 选手可使用快捷启动页面中的工具 selfEval 进行自测。在将答案文件（不必是全部题目）放到指定目录下后，即可选择全部或部分题目进行自测。注意：自测有次数限制，且自测结果仅用于选手调试，并不做为最终正式成绩。

贸易 (trade)

【题目描述】

近年来, A 国的商贸发展迅猛, 但国内的道路建设却跟不上步伐, 明显成为了人们贸易往来的限制, 管理者为此费尽了心思。

具体而言, A 国共有 $2^n - 1$ 个城市, 其中 1 号城市为首都。对于所有的非首都城市 i , 都有一条单向道路从城市 i 出发, 到达城市 $\lfloor \frac{i}{2} \rfloor$ 。为方便起见, 称这样的道路为“第一类道路”, 称城市 $\lfloor \frac{i}{2} \rfloor$ 为城市 i 的“上级城市”。

除此之外, 还有 m 条单向道路, 设其中第 i 条道路从城市 u_i 出发, 到达城市 v_i , 这样的道路都有一个特殊性质: 从城市 v_i 出发, 沿着第一类道路不断向“上级城市”走去, 最终总能走到城市 u_i 。称这样的道路为“第二类道路”。

每一条道路都有相应的长度值。由此, 对于 A 国的任意两个城市 x 和 y , 都可以计算出从城市 x 出发, 沿道路走到城市 y , 所经过的道路的长度之和的最小值, 将这一数值记为 $dist(x, y)$ 。但由于 A 国的道路建设存在严重缺陷, 从城市 x 出发可能根本到达不了城市 y , 方便起见, 此时定义 $dist(x, y) = 0$ 。同时一个城市出发到自己是不需要经过任何道路的, 因此定义 $dist(x, x) = 0$ 。

现在, 管理者希望计算出这些 $dist(x, y)$ 的值, 以便合理衡量人们贸易往来的便捷程度。但由于 A 国的城市数量太多, 将这些值一一列出的工作量太大, 因此管理者只希望求出所有 $dist(x, y)$ 值之和, 也就是 $\sum_{x=1}^{2^n-1} \sum_{y=1}^{2^n-1} dist(x, y)$, 并希望请你来帮忙。

【输入格式】

从文件 `trade.in` 中读入数据。

输入的第一行包含两个正整数 n 和 m 。

输入的第二行包含 $2^n - 2$ 个正整数, 第 i 个数 a_i 表示从城市 $i + 1$ 出发, 到达城市 $\lfloor \frac{i+1}{2} \rfloor$ 的“第一类道路”的长度。

接下来的 m 行, 每行包含三个正整数 u, v, w , 描述了一条从城市 u 到城市 v 的“第二类道路”, 其长度为 w 。

【输出格式】

输出到文件 `trade.out` 中。

输出一行一个正整数, 表示对应的答案。由于答案可能很大, 你只需要输出模 998244353 意义下的答案即可。

【样例 1 输入】

```

1 2 1
2 2 1
3 1 2 2

```

【样例 1 输出】

```

1 8

```

【样例 2】

见选手目录下的 *trade/trade2.in* 与 *trade/trade2.ans*。

【样例 3】

见选手目录下的 *trade/trade3.in* 与 *trade/trade3.ans*。

【样例 4】

见选手目录下的 *trade/trade4.in* 与 *trade/trade4.ans*。

【数据范围】

对于所有测试数据保证： $2 \leq n \leq 18$, $1 \leq m \leq 2^n$, $1 \leq u, v \leq 2^n - 1$, $1 \leq a_i, w \leq 10^9$ 。

测试点编号	n	m	是否有特殊性质
1 ~ 2	= 8	≤ 256	否
3 ~ 4	= 9	≤ 512	
5 ~ 8	= 12	$\leq 4,096$	
9	= 16	≤ 10	
10		≤ 50	
11		≤ 100	
12		$\leq 65,536$	是
13 ~ 15	否		
16 ~ 17	= 18	$\leq 262,144$	是
18 ~ 20			否

特殊性质：保证每一条“第二类道路”都是从首都（城市 1）出发。

字符串 (string)

【题目描述】

小 Y 是一名大学生, 最近正在研究字符串方向的问题。

小 Y 了解到关于字符串的如下定义:

- 给定一个长度为 n 的字符串 $s[1:n]$, 我们定义其子串 $s[l:r]$ ($1 \leq l \leq r \leq n$ 为选择 $s[l], s[l+1], \dots, s[r]$, 将其顺次拼接得到的新字符串。
- 给定一个长度为 n 的字符串 $s[1:n]$, 我们定义其翻转后的结果 $R(s)$ 为将 $s[n], s[n-1], \dots, s[1]$ 顺次拼接, 也就是将字符串反序拼接得到的字符串。
- 给定两个长度均为 n 的字符串 $a[1:n], b[1:n]$, 我们定义 a 的字典序小于 b 当且仅当存在 $1 \leq i \leq n$, 使得对于任意 $1 \leq j < i$, $a[j] = b[j]$, 且 $a[i] < b[i]$ 。

在了解了上述定义后, 小 Y 想到了这样的问题:

给定一个长度为 n 的字符串 $s[1:n]$ 。有 q 次询问, 每次询问给定两个参数 i, r 。你需要求出有多少 l , 满足如下条件:

- $1 \leq l \leq r$ 。
- $s[i:i+l-1]$ 字典序小于 $R(s[i+l:i+2l-1])$ 。

小 Y 想求助你帮忙解决这一问题。

【输入格式】

从文件 *string.in* 中读入数据。

本题有多组测试数据。

输入的第一行包含两个整数 c, t , 分别表示测试点编号和测试数据组数。 $c = 0$ 表示该测试点为样例。

接下来依次输入每组测试数据, 对于每组测试数据:

输入的第一行包含两个正整数 n, q , 表示字符串长度和询问次数。

输入的第二行包含一个长度为 n 的仅包含小写字母的字符串 s 。

输入的接下来 q 行, 每行包含两个正整数 i, r 。表示一次询问, 保证 $i + 2r - 1 \leq n$ 。

【输出格式】

输出到文件 *string.out* 中。

对于每一组测试数据的每一次询问, 输出一行一个整数, 表示满足条件的 l 的个数。

【样例 1 输入】

```
1 0 2
2 9 3
```

```

3 abacababa
4 1 4
5 2 4
6 3 3
7 9 3
8 abaabaaba
9 1 4
10 2 4
11 3 3

```

【样例 1 输出】

```

1 4
2 0
3 3
4 2
5 0
6 2

```

【样例 1 解释】

对于第一组数据的第一组询问：

- $l = 1$ 时, $s[i : i + l - 1] = \mathbf{a}$, $R(s[i + l : i + l + l - 1]) = \mathbf{b}$ 。
- $l = 2$ 时, $s[i : i + l - 1] = \mathbf{ab}$, $R(s[i + l : i + l + l - 1]) = \mathbf{ca}$ 。
- $l = 3$ 时, $s[i : i + l - 1] = \mathbf{aba}$, $R(s[i + l : i + l + l - 1]) = \mathbf{bac}$ 。
- $l = 4$ 时, $s[i : i + l - 1] = \mathbf{abac}$, $R(s[i + l : i + l + l - 1]) = \mathbf{baba}$ 。

这四种情况中, $s[i : i + l - 1]$ 的字典序均小于 $R(s[i + l : i + 2l - 1])$, 因此答案为 4。

【样例 2】

见选手目录下的 *string/string2.in* 与 *string/string2.ans*。

该样例数据范围满足测试点 5。

【样例 3】

见选手目录下的 *string/string3.in* 与 *string/string3.ans*。

【样例 4】

见选手目录下的 *string/string4.in* 与 *string/string4.ans*。

该样例数据范围满足测试点 24 ~ 25。

【数据范围】

对于所有测试数据保证： $1 \leq t \leq 5$ ， $1 \leq n \leq 10^5$ ， $1 \leq q \leq 10^5$ ， $1 \leq i + 2r - 1 \leq n$ ，字符串 s 仅包含小写字母。

测试点编号	n	q	特殊性质
1	≤ 5	≤ 5	A
2	≤ 10	≤ 10	
3	≤ 20	≤ 20	
4	≤ 50	≤ 50	
5	$\leq 10^2$	$\leq 10^2$	
6	$\leq 10^3$	$\leq 10^3$	无
7	$\leq 2,000$	$\leq 2,000$	
8	$\leq 3,000$	$\leq 3,000$	
9	$\leq 4,000$	$\leq 4,000$	
10	$\leq 23,333$	$\leq 23,333$	A
11	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	
12	$\leq 75,000$	$\leq 75,000$	
13	$\leq 9 \times 10^4$	$\leq 9 \times 10^4$	
14	$\leq 10^5$	$\leq 10^5$	
15	$\leq 23,333$	$\leq 23,333$	B
16	$\leq 75,000$	$\leq 75,000$	
17	$\leq 9 \times 10^4$	$\leq 9 \times 10^4$	
18	$\leq 10^5$	$\leq 10^5$	
19	$\leq 23,333$	$\leq 23,333$	无
20	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	
21	$\leq 75,000$	$\leq 75,000$	
22	$\leq 9 \times 10^4$	$\leq 9 \times 10^4$	
23	$\leq 95,000$	$\leq 95,000$	
24	$\leq 10^5$	$\leq 10^5$	
25			

- 特殊性质 A: 保证字符串中仅包含字符 **a** 和 **b**，且每个字符独立等概率的在 **a** 和 **b** 中选择。
- 特殊性质 B: 保证字符串中的相邻字符互不相同。

合并书本 (book)

【题目描述】

小 C 有 n 本书，每本书都有一个重量，他决定把它们合并成一摞。

每一次合并小 C 可以把一摞书放到另一摞书上面，使得它们合并到一摞。如果小 C 把第 i 摞书放到第 j 摞书上面，小 C 需要消耗的体力为第 i 摞书的重量加上两摞书的磨损值之和。

初始时每本书自成一摞且磨损值均为 0。每当小 C 将两摞书合并后，形成的新的一摞书的磨损值为合并前的两摞书的磨损值的较大值的两倍再加一，重量为合并前的两摞书的重量之和。

你的任务是设计出合并的次序方案，使小 C 耗费的体力最少，并输出这个最小的体力耗费值。

【输入格式】

从文件 `book.in` 中读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数 t ，表示数据组数。

接下来依次输入每组测试数据，对于每组测试数据：

输入的第一行包含一个正整数 n ，表示有 n 本书。

输入的第二行包含 n 个正整数，第 i 个数 w_i 表示第 i 本书的重量。

【输出格式】

输出到文件 `book.out` 中。

对于每组测试数据输出一行一个整数，表示将 n 本书合并成一摞需要消耗的最少体力。

【样例 1 输入】

```
1 1
2 4
3 1 1 1 1
```

【样例 1 输出】

```
1 6
```

【样例 1 解释】

如果小 C 将 4 本书两两合并再将得到的两摞合并成一摞。

那么前两次需要消耗的体力各为 1，第三次将一摞重量为 2 的书放到另一摞上面，两摞书磨损值各为 1，需要消耗的体力为 $2 + 1 + 1 = 4$ 。

因此如果选择这个方案，小 C 耗费的体力只有 $1 + 1 + 4 = 6$ 。

可以证明，在上述例子中，6 为最小的体力耗费值。

【样例 2】

见选手目录下的 *book/book2.in* 与 *book/book2.ans*。

【样例 3】

见选手目录下的 *book/book3.in* 与 *book/book3.ans*。

【样例 4】

见选手目录下的 *book/book4.in* 与 *book/book4.ans*。

【数据规模和约定】

对于所有测试数据保证： $1 \leq t \leq 10$ ， $1 \leq n \leq 100$ ， $1 \leq w_i \leq 10^9$ 。

测试点编号	$n \leq$	是否有特殊性质
1 ~ 2	7	否
3	11	
4	13	
5 ~ 6	22	
7 ~ 8	28	
9 ~ 13	50	
14	60	
15	70	
16	80	
17 ~ 18	100	是
19 ~ 20		否

特殊性质：保证 $w_i = 1$ 。