

Parliament

Zhenhua Lin

August 2010

1 Problem

Description New convocation of The Fool Land's Parliament consists of N delegates. According to the present regulation delegates should be divided into disjoint groups of different sizes and every day each group has to send one delegate to the conciliatory committee. The composition of the conciliatory committee should be different each day. The Parliament works only while this can be accomplished. You are to write a program that will determine how many delegates should contain each group in order for Parliament to work as long as possible. (source: Northeastern Europe 1998)

Input The input file contains a single integer N ($5 \leq N \leq 1000$).

Output Write to the output file the sizes of groups that allow the Parliament to work for the maximal possible time. These sizes should be printed on a single line in ascending order and should be separated by spaces.

Sample Input 7

Sample Output 3 4

2 Solution

Before introduce the solution, let us review an inequality and the multiplication principle described below.

Theorem 1 (Inequality of Arithmetic and Geometric Means). *For any list of n nonnegative numbers x_1, x_2, \dots, x_n , we have*

$$x_1 x_2 \dots x_n \leq \left(\frac{x_1 + x_2 + \dots + x_n}{n} \right)^n$$

and the equality holds if and only if $x_1 = x_2 = \dots = x_n$.

Theorem 2 (Multiplication Principle). *If a task consists of k different operations o_1, o_2, \dots, o_k , and each operation o_i can be done by m_i ways. Then, there are in total $m_1 m_2 \dots m_k$ different ways to complete the task.*

Back to our problem on hand. Assume that all delegates are separated into k disjoint groups, each of which has x_i delegates. Then, according to multiplication principle, there are $M = x_1 x_2 \dots x_k$ different ways to compose a conciliatory committee. According to the inequality of arithmetic and geometric means, M reaches its maximal value when $x_1 = x_2 = \dots = x_k$.

Unfortunately, for any x_i and x_j , if $i \neq j$, then $x_i \neq x_j$, and x_i should be an integer. However, the inequality of arithmetic and geometric means does give us some intuition: M should be as larger as x_1, x_2, \dots, x_k get close to each other. The following lemmas confirms this intuition. Without loss of generality, assume $x_1 < x_2 < \dots < x_k$. Define the *gap* of two integers x_1 and x_2 as the number of integers between them. Denote it by $gap(x_1, x_2)$. For example, $gap(5, 8) = 2$ since there are 6 and 7 between them. If $gap(x_1, x_2) = 0$, then we say that x_1 and x_2 has *no gap*.

Lemma 1. *Let $t = x_1 + x_2$ when t, x_1 and x_2 are nonnegative integers and $x_1 \neq x_2$. $x_1 x_2$ has the maximal value if and only if $gap(x_1, x_2) \leq 1$. Particularly, when $t \geq 5$, the maximum of $x_1 x_2$ is larger than t strictly.*

Given a increasing sequence of integers $X_k = x_1, x_2, \dots, x_k$, define the number of gaps in X_k as the number of pairs (x_i, x_{i+1}) which $gap(x_i, x_{i+1}) > 0$. For example, sequence 1, 2, 5, 6, 8 has 2 gaps: one between 2 and 5, and the other between 6 and 8.

Lemma 2. *M reaches the maximal value if and only if there is at most one gap in X_k and that gap is at most 1 if any.*

The lemma above tells us that to make M as large as possible, the sequence X_k should composed by a list of continuous integers $x + 1, x + 2, \dots, x + k$, or two segments of continuous integers $x + 1, x + 2, \dots, x + u$ and $(x + u + 1) + 1, (x + u + 1) + 2, \dots, (x + u + 1) + v$ where $u + v = k$.

Example $N = 15, k = 3$. Then $M = 4 \times 5 \times 6 = 120$. If $k = 4$, then $M = 2 \times 3 \times 6 = 144$. For $k = 5, M = 1 \times 2 \times 3 \times 4 \times 5 = 120$.

One natural question is that, given k is fixed, how to look for valid sequence X_k which maximize M ? Well, by lemmas above, we can assume that $X_k = x + 1, x + 2, \dots, x + u, (x + u + 1) + 1, (x + u + 1) + 2, \dots, (x + u + 1) + v$ where $u + v = k$ and $v < k$. So

$$\left(\sum_{i=1}^u x + i\right) + \left(\sum_{j=1}^v (x + u + 1) + j\right) = N$$

Or

$$x = \frac{N}{k} - \frac{v}{k} - \frac{k+1}{2}$$

by assuming $N = mk + t$ where $t < k$,

$$x = m - \frac{t - v}{k} - \frac{k + 1}{2}$$

Since x is integer and $v < k$, when k is odd, $v = t$. When k is even, $v = t \pm \frac{k}{2}$, depending on if t is larger than half of k . By this way, for any N and valid k , we compute the unique pair of x and v .

Lemma 3. *Given N and k , the sequence X_k maximizing M is unique.*

At this point, we can solve the problem as follows: For all possible k , compute the x and v by above way. Then compute their corresponding M values, between which we pick the largest one. The algorithm is inefficient since we involve computing the pretty number of many factors (imagine $4 \times 5 \times \dots \times 50$!!). Java and some programming language do provide BigInteger class, but it is time consuming.

The question following is, can we void manipulation on big integers? Yes. To make it, we need some insightful thoughts. From the example given above, it seems that M increases and x_1 decreases when k get larger, but not too large. At least, $x_1 = 1$ is not expected and should be voided. In a word, we try to get an $x_1 \geq 2$ as small as possible. The perfect value for x_1 would be 2, of course. However, not always we can make it. The good news is, x_1 should be 2 or 3.

Lemma 4. *Let X_k be the sequence maximize M among all valid sequences. Then x_1 is 2 or 3.*

Proof. We prove it by contradiction. Assume $x_1 > 3$. Consider the case $x_1 = 4$. By Lemma 2, $x_2 = 5$ or 6. If $x_2 = 5$, we construct a new sequence X' by replacing x_2 with $x'_1 = 2$ and $x'_2 = 3$ and keeping others intacted. Note that X' is a valid sequence, meaning that all elements in it are different and sum to N . Then $\frac{X_k}{X'} = \frac{5}{6}$. That's $X_k < X'$, which is a contradiction. If $x_2 = 6$, we construct a new valid X' by replacing x_1 and x_2 by 2,3 and 5. Since $2 \times 3 \times 5 = 30 > 4 \times 6 = 24$, it contradicts the optimality of X_k . Consider $x_1 > 4$. In this case, we construct the new valid X' by replacing x_1 with two small factors y_1 and y_2 such that $gap(y_1, y_2) \leq 1$. According to Lemma 3, $X' > X_k$, yet another contradiction. So $x_1 \leq 3$. Obviously, if $x_1 = 1$, we can easily construct an valid $X' > X_k$.

Now, we narrow the search scope to those X_k starting with 2 or 3. Furthermore,

Lemma 5. *Let $X = x + 1, x + 2, \dots, x + u_1, (x + u_1 + 1) + 1, (x + u_1 + 1) + 2, \dots, (x + u_1 + 1) + v_1$ be the sequence maximizing M starting from 2 ($x = 1$) and $Y = y + 1, y + 2, \dots, y + u_1, (y + u_2 + 1) + 1, (y + u_2 + 1) + 2, \dots, (y + u_2 + 1) + v_2$ starting from 3 ($y = 2$), where $u_1, u_2 \geq 1$ and $v_1, v_2 \geq 0$. Let i be the length of X and j be the length of Y . Then, $i > j$ and $v_1 < v_2$.*

Proof. Obviously, $v_1 < i$. $\sum_{a \in X} a = N \Rightarrow (2i + 3)^2 = 8N - 8v_1 + 9$ and $\sum_{a \in Y} a = N \Rightarrow (2j + 5)^2 = 8N - 8v_2 + 25$. Combining them, we have

$$(j - i + 1)(i + j + 4) = 2(v_1 - v_2) + 4$$

Assume $i \leq j$. Then $2(v_1 - v_2) + 4 \geq i + j + 4 \geq 2i + 4$, or $v_1 - v_2 \geq i$. However, since $v_1 < i$ and $v_2 \geq 0$, it's impossible. So $i > j$ and then $v_1 < v_2$.

The above lemma simply states that if we get two valid sequences, namely, one starting from 2 with length i which maximize M when $k = i$, and the other starting from 3 with length j which maximize M when $k = j$, and each of them consists of two segments of contiguous integers, then latter segment of the one starting from 2 is short than that of the one starting from 3.

This fact leads to following important lemma.

Lemma 6. *Let valid $X = x + 1, x + 2, \dots, x + u, (x + u + 1) + 1, (x + u + 1) + 2, \dots, (x + u + 1) + v$ start from 3 ($x = 2$). Then X maximizes M among all valid sequences if and only if $v = 0$ or $v = 1$.*

Proof. In either case of $v = 0$ or $v = 1$, no valid sequence starting from 2 can be constructed according Lemma 5. According to Lemma 4, X maximizes M . Now assume X maximize M but $v > 1$. Then, we construct a new valid Y by replacing $(x + u + 1) + 2$ with 2 and $(x + u + 1)$. Since $u > 0$ and $x = 2$, $2(x + u + 1) > (x + u + 1) + 2$. So $Y > X$. Contradiction.

Now we come to the crux of the problem. We construct a valid sequence X starting from $x + 1$ where $x = 1$ by the following way: start from $x + 1$, we keep adding $x + 2, \dots, x + i$ until adding $x + i + 1$ will make the sum of elements in X larger than N . Then, we increases each element by 1 in the order of $x + i, x + i - 1, \dots, x + 1$ and repeat the process until it sums to N . The sequence we finally get is the answer.

The next step is to prove the algorithm is correct. Given above lemmas, it's an easy task.

Proof. Firstly, $m \leq k + 1$. Otherwise, the first while loop won't terminate. So the body of second while loop will be executed at most $k + 1$ times. If $m = k$ or $m = k + 1$, then X starts from 3 with the second segment of length 0 or 1, respectively. According to Lemma 6, X is optimal. If $m = 0$, we are unable to construct a sequence Y such that Y starts from 3 and will the second segment of length at most 1. If $0 < m < k$, X has second segment of length at least 1. According to Lemma 5, we also can't construct Y maximizing M and starting from 3 with second segment of length at most 1. Combining Lemma 4, the X is optimal.

```
1:  $S \leftarrow 0$ 
2:  $k \leftarrow 0$ 
3: array  $X$  stores the sequence
4: while  $S + (k + 2) \leq N$  do
5:    $X[k + 1] \leftarrow k + 2$ 
6:    $k \leftarrow k + 1$ 
7:    $S \leftarrow S + X[k]$ 
8: end while
9:  $j \leftarrow k$ 
10:  $m \leftarrow N - S$ 
11: while  $m > 0$  do
12:    $X[j] \leftarrow X[j] + 1$ 
13:    $m \leftarrow m - 1$ 
14:    $j \leftarrow j - 1$ 
15:   if  $j < 1$  then
16:      $j = k$ 
17:   end if
18: end while
```
