

Sắp xếp từ

Có N từ được đánh số từ 1 tới N , từ thứ i có độ dài là $\text{length}(i)$.

Một cách sắp xếp N từ vào K dòng được biểu diễn bởi bộ $K - 1$ số $(A_1, A_2, \dots, A_{K-1})$, $1 \leq A_1 < A_2 < \dots < A_{K-1} < N$, với ý nghĩa các từ từ 1 đến A_1 sẽ nằm ở dòng 1, các từ từ $A_1 + 1$ đến A_2 sẽ nằm ở dòng 2, cứ như thế và cuối cùng các từ từ $A_{K-1} + 1$ đến N sẽ nằm ở dòng K - dòng cuối cùng.

Trên mỗi dòng, giữa 2 từ liên tiếp phải có đúng 1 dấu cách giữa chúng, do đó nếu ký hiệu $\text{line}(w)$ là độ dài của dòng thứ w , và dòng w chứa các từ từ i đến j thì dễ thấy ta có công thức:

$$\text{line}(w) = \text{length}(i) + \text{length}(i+1) + \dots + \text{length}(j) + (j - i)$$

trong đó $(j - i)$ là số dấu cách.

Độ "mượt" của 1 cách sắp xếp N từ vào K dòng được tính bởi công thức:

$$S = |\text{line}(1) - \text{line}(2)| + |\text{line}(2) - \text{line}(3)| + \dots + |\text{line}(K - 1) - \text{line}(K)|$$

S càng nhỏ ta nói cách sắp xếp càng đẹp.

Yêu cầu: Bạn được cho biết độ dài N từ, hãy tìm một cách sắp xếp N từ vào K dòng với K là một số **tùy ý** thuộc khoảng $[1, N]$ sao cho độ "mượt" S là nhỏ nhất có thể. Cần lưu ý thêm, độ dài của mỗi dòng không thể quá dài, $\text{line}(w)$ phải luôn không quá M với M cho trước.

Input: đọc từ file **arrange.in**

Dòng đầu tiên chứa số nguyên dương T là số lượng bộ dữ liệu. T nhóm dòng sau, mỗi nhóm dòng mô tả một bộ dữ liệu theo định dạng:

- Dòng đầu gồm 2 số nguyên dương M và N ($M \leq 10^6$).
- Dòng thứ hai chứa N số nguyên dương $\text{length}(1), \text{length}(2), \dots, \text{length}(N)$ ($\text{length}(i) \leq M$ với mọi i).

Output: ghi ra file **arrange.out**

In ra giá trị nhỏ nhất có thể của độ "mượt" S trên một dòng.

Subtask:

Subtask 1 (20%): $T = 1, N \leq 20$

Subtask 2 (30%): $T = 1, N \leq 500$

Subtask 3 (40%): $T = 1, N \leq 2000$

Subtask 4 (10%): $T \leq 50, N \leq 2000$

Ví dụ 1:

arrange.in	arrange.out
1 6 4 4 3 2 5	3

Ví dụ 2:

arrange.in	arrange.out
1 4 2 1 2	0