

Giao thức truyền thông

Anh Halfnet làm việc ở công ty truyền thông Bytetel và là nhân viên thiết kế các giao thức truyền thông trên mạng. Các máy tính đ-ợc nối với nhau bằng cáp truyền thông, qua đó, một máy có thể truyền dữ liệu cho máy khác đ-ới các tín hiệu ở k mức điện thế khác nhau. Tuy vậy điện thế chỉ có thể thay đổi sau mỗi $1/n$ giây (trong khoảng thời gian $1/n$ giây điện thế không thay đổi và đ-ợc gọi là một *xung*). Dữ liệu đ-ợc gửi đ-ới dạng từng gói gồm m xung liên tiếp (tức là cần m/n giây để gửi một gói tin).

Vì các lý do kỹ thuật, điện áp phải không cố định trong thời gian truyền một gói tin, mà luôn thay đổi. Nói một cách chính xác, một máy không thể truyền gói tin gồm l xung liên tiếp ở cùng một mức điện thế.

Nếu giao thức cho phép gửi x gói tin khác nhau, ta nói có thể mã hoá $\log_2 x$ bit thông tin trong một gói. Halfnet tự hỏi, một máy tính tối đa có thể truyền đi bao nhiêu bit trong một giây.

Ví dụ: Giả thiết cáp có thể truyền tải tín hiệu ở hai mức điện thế ($k=2$), ký hiệu là 0 và 1. Nếu điện thế có thể thay đổi 20 lần trong một giây ($n=20$), gói tin gồm 4 xung ($m=4$), trong đó không có 3 xung liên tiếp cùng một mức điện thế ($l=3$), thì khi đó không thể gửi các gói tin 0000, 0001, 1000, 1111, 1110, 0111, nh-ng có thể gửi đ-ợc các gói tin 0010, 0011, 0100, 0110, 0101, 1101, 1100, 1011, 1001 và 1010. Nh- vậy có thể gửi đ-ợc 10 gói tin khác nhau, điều đó cho phép mã hoá $\log_2 10$ bit thông tin trong mỗi gói. Trong vòng 1 giây có thể truyền $20/4=5$ gói tin, có nghĩa là $5 \cdot \log_2 10 \approx 16.6096$ bit thông tin.

Yêu cầu: Tính số gói tin khác nhau và số bit tối đa có thể truyền trong 1 giây

Dữ liệu: Vào từ file văn bản PROT.INP gồm một dòng 4 số nguyên k, n, m, l ($2 \leq k \leq 10, 1 \leq n \leq 1000, 1 \leq m \leq 100, 2 \leq l \leq m$), ngoài ra còn có điều kiện $n/m \leq 10, n:m$.

Kết quả: Đ- ra ra file văn bản PROT.OUT hai số nguyên - số gói tin khác nhau và số bit tối đa có thể truyền trong 1 giây (làm tròn xuống tới số nguyên gần nhất)

Ví dụ:

PROT.INP
2 20 4 3

PROT.OUT
10 16