Que onda chavos la neta no tengo ni la más mínima idea de quien es cada ejercicio, pero bueno ahí buscan el suyo, al final de cuentas por lo que entendí solo hay que ofrecer una solución general, y por que se escogió ese método o ¿no es así?, así que espero esté bien.

**Líneas que se cortan:**

Representamos los puntos como ecuaciones parametricas, Recordemos que cualquier punto en un segmento de línea ab puede ser escrito como una combinación convexa empleando un parámetro real s:

p(s) = (1 - s)a + sb Para 0 <= s <= 1

De forma similar para cd podemos introducir un parámetro t:

q(t) = (1 - t)c + td para 0 <= t <= 1

Una intersección occurre si y sólo si podemos encontrar valores

para s y t en los rangos deseados tales que p(s) = q(t)

Por lo tanto obtenemos las dos ecuaciones siguientes:

(1 - s)ax + sbx = (1 - t)cx + tdx

(1 - s)ay + sby = (1 - t)cy + tdy

El cálculo de s y t implica una división

Si el divisor es cero, significa que los segmentos de línea son paralelos o colineales

Estos casos especiales deben ser tratados con cuidado Si el divisor es distinto de cero, entonces obtendremos valores para s y t como número racionales.

**Text Formatting**

Para este ejercicio solo es separar de acuerdo al numero de caracteres quieren que mostremos, lo que hay que ir haciendo es ir comparando la cadena, y llevar un contador, si nuestra cadena es mayor al numero de caracteres máximos, entonces la pasamos a la siguiente línea si es menor la escribimos, y claro para leer los archivos y separarlos en java la función “tokenizer”. Un método que haya escogido no sabría indicarles pero esa sería una posible solución.

**MBone**

Para este ejercicio que nos habla sobre los routers y puertos de host y no se cuantas cosas más, vamos a hacerlo con grafos (graph), hay muchas maneras de representar los grafos en programación sea en el lenguaje que sea, en nuestro programa buscaríamos mediante grafos la ruta más corta del punto x al punto y, en este caso del router origen al router destino, buscando todos los caminos posibles de acuerdo a los tipos de conexiones que haya entre ellos (tipos de redes; bus, estrella, anillo, etc.), este método de grafos es el indicado ya que si lo hacemos con matrices mientras más routers tengamos más lento será nuestro programa en forma exponencial.

**Robotic Sort**

Para este ejercicio identificaríamos primero los tamaños y buscamos el más pequeño y lo rotamos a la primera posición obviamente con sus auxiliares para ir moviendo los demás tubos o muestras, después buscamos el segundo y hacemos lo mismo movemos todo lo que este detrás del primero y antes del segundo y rotamos, y así sucesivamente aunque pondríamos dos métodos en caso de ser mayor la siguiente muestra, llevar un contador y cuando se encuentre una muestra mayor detener el contador y rotar el numero de muestras que tenga contador. Y Si la muestra es menor no hacer nada y seguir buscando ya que estará ordenada.

**Weird Numbers**

Para dar una solución óptima tendremos que crear un método que se encargue de hacer la conversión de binario y/o binario negativo a decimal y viceversa y para leer las líneas del archivo de entrada con un método tokenizer, ya solo habrá que buscar un método para convertir de negativos a enteros por que el de las hojas no lo entendí del todo, pero sería que le explicaras ese método que viene en las hojas y ya con ese método se resolvería.

**Perfección**

Para este programa utilizaremos varias sentencias de repetición y varios condicionales y algunos contadores, para ir checando si es divisor del numero x, en caso de serlo sumarlo al final si la suma obtenida es igual al número x entonces es perfecto, en cambio si se pasa es abundante en ultimo caso es deficiente.