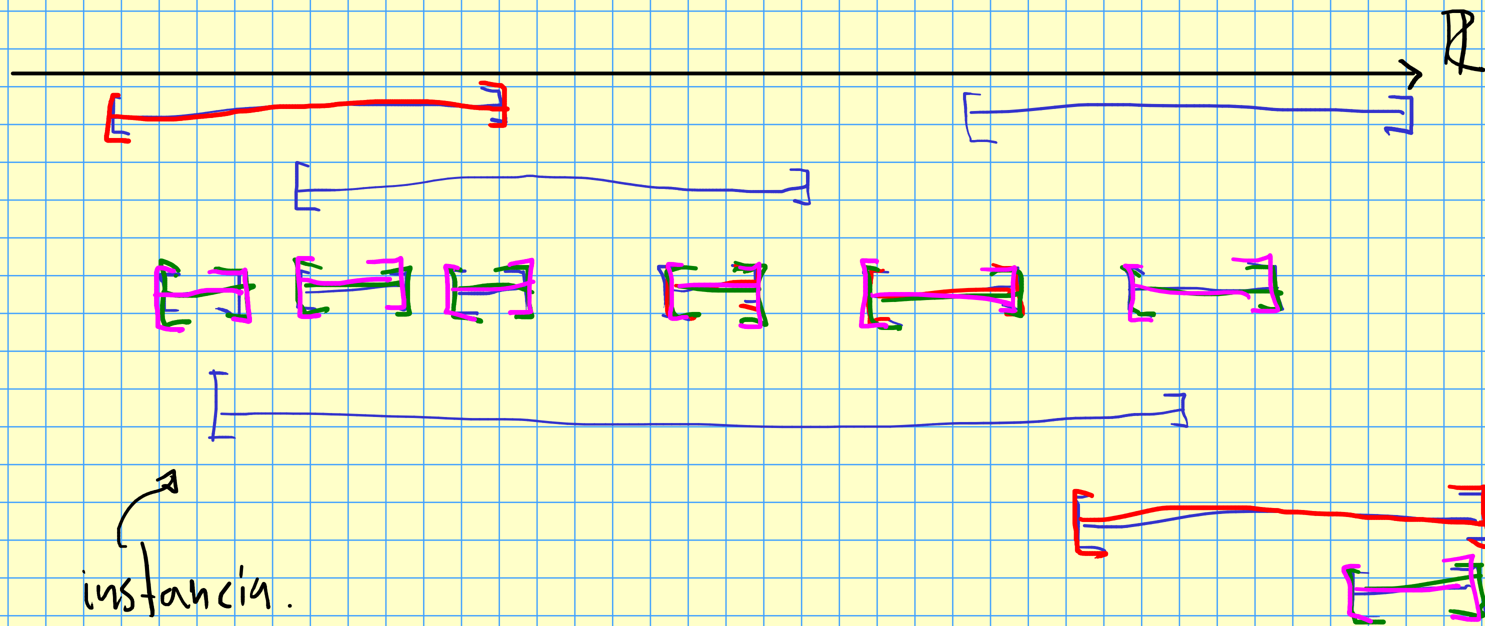


4 películas.

6 películas

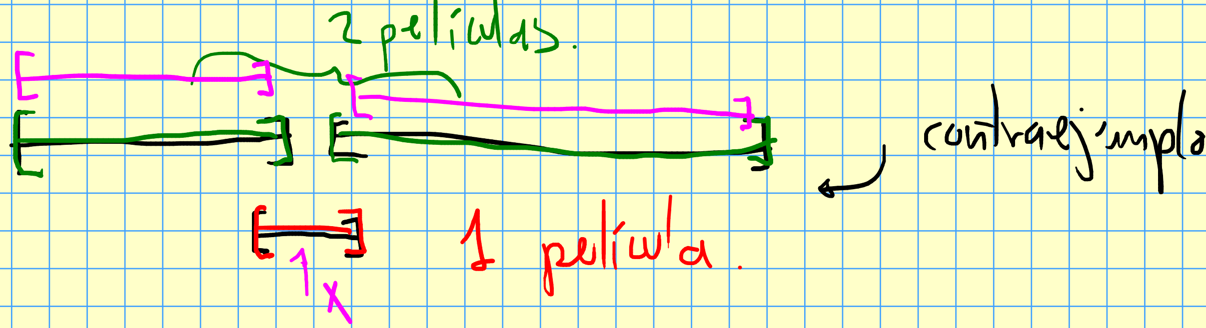
Ordenar por cuando comienzan.

Posible algoritmo: Ver la primera que podamos, luego la siguiente que podamos y así sucesivamente.

Entrada.

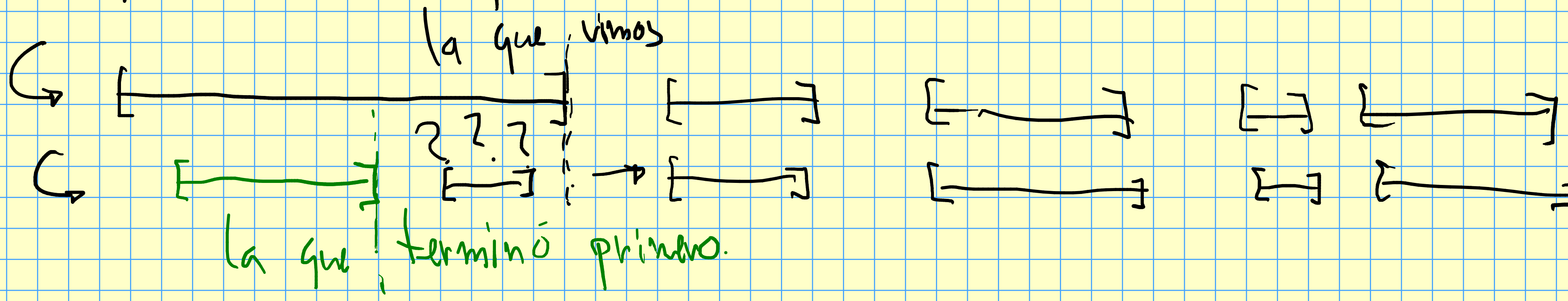
$$[a_1, b_1]$$
$$[a_2, b_2]$$
$$[a_3, b_3]$$
$$\vdots$$
$$[a_n, b_n]$$

Posible algoritmo 2 Ordenarlas por duración y ver de las que duran menos, a las que duran más las más posibles.



Posible algoritmo 3 Ordenarlas por su hora en que terminan.
De entre las películas que haya, siempre veremos aquella que termine primero.

Pensemos en la primer película que veremos. Sup. que **no** fue la que terminó primero.



Esto nos lleva ~~un~~ una nueva forma de ver al menos tantas películas como vimos y quizás más. Por lo tanto S.P.G. la primer película que vimos es la que terminó primero.

```
Incrementar(y):
    si y=0, respondemos 1, si no:
        si y es impar, entonces
            respondemos 2*Incrementar(Parte-entera(y/2))
        si y es par, respondemos(y+1)
```

Vamos a demostrar que es correcto por inducción en n.

<u>BI</u>. Si $n=0$, el algoritmo responde 1, lo cual es correcto.

<u>HI</u>. Sup. que para todo $k<n$ el algoritmo da la respuesta correcta.

<u>PI</u>. Para n da la respuesta correcta.

- Si n es par el algoritmo responde $n+1$, lo cual es correcto.
- Si n es impar el algoritmo responde:

$$2\,\text{Incrementar}\left(\left\lfloor \frac{n}{2} \right\rfloor\right) = 2\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) = 2\left(\frac{n-1}{2}\right) + 2 = n-1+2 = n+1.$$

Así, el PI se cumple y por lo tanto el algoritmo es correcto siempre.

$\square$