

Situación

↙

2	-1	3	7	0	-3	-5	-7	8	9
---	----	---	---	---	----	----	----	---	---

Algoritmo específico

A priori se ve necesaria una función que sume n elementos tras dos índices dados:
 $n = \text{tamaño del vector} - 1$

```
int suma (int *v, int ini=0, int fin=n) { O(n)
    int suma;
    for (desde inicio hasta fin)
        suma += v[i];
    return suma;
}
```

Como estructura auxiliar se usará un set de series donde tenemos la siguiente estructura

```
struct serie {
    int ini, fin, suma;
}
```

La ordenación de estos structs viene dada por la función `ordenaSeries` de orden $O(1)$

```
bool OrdenaSeries (serie1, serie2) { O(1)
    return serie1.suma < serie2.suma;
}
```

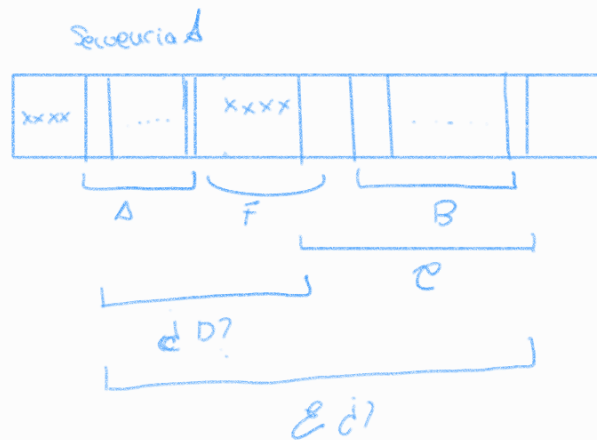
La idea es la siguiente, recopilar en el set todas las series (matarmoscas a cañonazo) o usar otra estructura de manera que si la suma es mayor se cambia.

Se usarán punteros para determinar todos y cada uno de los límites posibles a sumar ocasionando un algoritmo lineal-logarítmico o cuadrático → cuestión de implementar.

Divide y vencerás



Esa sería una posible división de manera que el pegado sólo consistiría en elegir el máximo de unas secuencias que dependen de los límites de los dos unidades:



Duo hay que considerarlo
pues si añadimos términos
yo no sería lo mayor \Rightarrow
obligatoriamente sería un
 \hookrightarrow si $F=0 \Rightarrow$ sí se considera

Para E encontramos la
misma casística.

¿Casística sustancial?