1: Explica como si tuviera 5 años

a) búsqueda lineal.

- Imagina una montaña así  Estás hasta arriba y tienes una pelota a control remoto; Esa pelota la vas a aventar, & quieres que llegue ~~hasta abajo~~ al punto más bajo de la montana entonces vas a controlar la pelota para que no se quede atorada en uno de los "valles" antes del fondo (Nota; controlas la caída pero la pelota no puede subir laderas), => siempre vas a direccionar la pelota hacia una trayectoria que siga bajando (p) con pasos de tamaño variante (α)

b) Región de confianza.

- Imagínate que tienes una pelota de velcro, una base en el que el velcro se pega y tienes los ojos tapados.

Así 

Estás intentando poner la pelota en el punto más profundo de la base (el rojo) & tienes un amigo que te dice si en tu último intento, la pelota bajó mas, se quedó igual o subió. Naturalmente, lo que vas a hacer ~~es escoger una direccion en la que~~ empezarla a soltar ~~en~~ en ~~u~~ ca regiones en las que crees que decienda & cuando ya no ~~bajes~~ llegue más profunda la pelota en esa región vas a intentar hacer menores ~~cambios de~~

Cambios de ~~preste~~ posic de tu mano para que la pelota caiga cada vez más abajo

2) $f(x) = \frac{1}{2} x^T Q x - b^T x$, Q semi-pos-def. queremos el minimizador de 1 dimensión s/ la línea $x_k + \alpha p_k$.

→ Notamos que $\nabla f(x) = Qx - b$ y suponiendo que $\exists!\ Qx = b$.

$$\varphi(\alpha) = f(x_k + \alpha p_k) = \frac{1}{2}(x_k + \alpha p_k)^T Q (x_k + \alpha p_k) - b^T (x_k + \alpha p_k)$$

de
derivando φ c.r.a α y igualando a cero nos queda

$$\cancel{\varphi'(\alpha) = \frac{1}{2}(p_k^T Q x_k + x_k^T Q p_k) - b^T p_k = 0}$$

$\Leftrightarrow$

$$\varphi'(\alpha) = (Q(x_k + \alpha p_k) + b)^T p_k = 0$$

$$\Leftrightarrow \quad \overbrace{(Qx + b)^T}^{\nabla f_k} p_k + \alpha p_k^T Q p_k = 0$$

$$\Leftrightarrow \quad \nabla f_k^T p_k = - \alpha p_k^T Q p_k$$

$$\Leftrightarrow \quad \alpha = - \frac{\nabla f_k^T p_k}{p_k^T Q p_k}$$

α es el minimizador

3) ~~Así esta~~

Dado que tenemos más puntos que cámaras $\Rightarrow$ no podemos poner una cámara en cada lugar donde sucedió un crimen. Lo que propongo es lo siguiente para posidar las 8000 cámaras. Vamos a primero meter las coord. al intervalo $[0,1]$ usando: $(a,b) \to (0,1) \Rightarrow f(x) = \frac{1}{b-a}(x-a)$

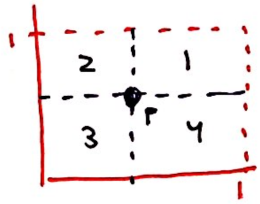
y así trabajaremos.

1: Usar Newton Modificado y encontrar el punto optimo c.r.a las distancias; esto es:

$$\min_{x,y} \sum_{i=1}^{n} \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

y eso nos dará un punto en el intervalo $(0,1)\times(0,1)$. [lo llamamos p]

2: Tomar p y dividir a los puntos del Dataset en 4 regiones; así $\longrightarrow$ tomando a p. como "centro".



Ahora a los puntos en ~~2~~ 1,2,3,4 los minimizamos usando el mismo método mencionado y tomamos P_1', P_2', P_3', P_4'.

3) repetir el procedimiento hasta tener 8,000 p_i's. Las p_i serán los puntos donde se colocarán las cámaras.

- El razonamiento es: debemos dividir al cuadro unitario en puntos pero las "densidades" de crimen son distintas, ~~en~~ no uniformes =>

1) Un mallado uniforme no sirve pues ~~al~~ no vas a poner camaras en el bosque

2) Un mallado uniforme + optimizar la dist. minima de cada punto de la malla ~~no sirve~~ a los k crímenes ++ cercanos no sirve; vas a terminar delineando la ciudad.

3) Tirar puntos aleatoreas en el cuadro & sacar $\sum$ de sus distancias a cada crimen No sirve, ~~que dirección de dece~~ ¿cómo vas a optimizar puntos aleatorios?

Nota: Correr el algoritmo entregado es pesado, va a tardar algunas horas.