Problema 1

Se leen las chapas en un arreglo. Luego, para cada llave, se realiza una búsqueda binaria en el arreglo de chapas y se regresa el lugar de la chapa correspondiente (los lugares empiezan desde 1). En caso de no existir, se regresa 0.

Problema 2

Podemos pre-computar un arreglo (ordenado) con todos los números de Fibonacci que quepan en el rango de un número unsigned long long. Hay que tener cuidado con el desbordamiento de enteros en este caso. Ya teniendo el arreglo y el índice máximo del mayor número de Fibonacci que cabe en un unsigned long long, podemos hacer una búsqueda binaria para determinar el índice de un número de Fibonacci dado.

Problema 3

Se puede pre-computar un arreglo con todas las sumas consecutivas de x_i. Por construcción, este arreglo está ordenado. Para cada pregunta, se hace una búsqueda binaria modificada que regrese el mayor elemento menor a la cantidad de gasolina dada y se regresa la diferencia.

Problema 4

Obviamente las posiciones deben estar ordenadas, de otra forma no tenemos ninguna propiedad que nos ayude a resolver el problema. Luego, sea d la mínima distancia entre programadores. Nos hacemos la siguiente pregunta: Dado un valor deseado de d, ¿es posible ubicar P programadores en el arreglo ordenado dado tal que la distancia mínima entre ellos sea d? Supongamos que podemos resolver este problema más pequeño, ¿cómo resolveríamos el problema original?. Es fácil notar que el mayor valor de d es la distancia D entre la primera y la última posición ya que son los dos valores más lejanos. Para resolver el problema original, podemos iterar desde $d = 1$, $d = 2$, ... ,$d = D$ y preguntarnos ¿es posible ubicar P programadores en el arreglo ordenado dado tal que la distancia mínima entre ellos sea d?. En algún momento no va a ser posible ubicar P programadores bajo una mínima distancia d. Al llegar a este punto, simplemente retornamos el último valor de d donde sí fue posible. Este método no es más que una búsqueda lineal sobre los posibles valores de d. Para realizarlo de manera óptima, podemos hacer búsqueda binaria sobre los valores de d: si la respuesta a nuestro sub-problema fue "Si es posible", nos movemos hacia la derecha, en caso contrario nos movemos hacia la izquierda (tal como en una búsqueda binaria normal).

Ahora vamos a resolver el sub-problema que planteamos. Para $d = 1$ se pueden ubicar N programadores y por lo tanto si es posible ya que $P \leq N$. ¿Cómo generalizamos para valores arbitrarios de d? Si empezamos asignando el primer programador a la primera posición (la menor ya que ordenamos el arreglo previamente), la siguiente posición posible para un programador es la posición más cercana que exceda en d posiciones a esta primera posición. De esta forma podemos generar un ejemplo que cumpla que la distancia mínima es d. Dicho ejemplo dará valor de $P^{'}$ programadores. Si $P^{'} \geq P$ entonces este ejemplo nos asegura que si es posible ubicar P programadores ya que ubicamos a más programadores. Sin embargo, si $P < P^{'}$, no puede existir ninguna ubicación de P programadores con mínima distancia d. La demostración de este hecho queda como ejercicio al lector, pero al menos debe de ser intuitiva por la construcción de la ubicación con $P^{'}$ programadores.

Problema 5

Ver el vídeo de la sesión para la explicación.