Problemas de Diseño y Análisis de Algoritmos

Olivia González Peña
Juan Carlos Casteleiro Wong
Adrian Tubal Páez Ruiz

2018 - 2019

Problema 1: Alumbrando Calles

Para llegar a su casa Fito debe pasar por una calle muy larga de longitud n metros, que está en una zona algo apartada. Cuando es de noche, la calle es iluminada usando unos postes de luz, que están distribuidos de cierta forma a lo largo de uno de los lados de la calle. En noches muy oscuras, por falta de luna, esa calle se vuelve un poco peligrosa por la falta de bombillos en los postes. Para solucionar el problema de la iluminación se decide poner bombillos en estos, buscando iluminar de noche toda la calle. Como el CDR no tiene mucho presupuesto se quiere determinar cuál es la menor cantidad de bombillos que es necesario comprar. Un bombillo, una vez puesto en su poste, es capaz de iluminar la calle m metros hacia los dos lados, es decir que si ponemos un bombillo a 7 metros del inicio de la calle y el rango de iluminación es de 3 metros entonces ese bombillo iluminará desde 4 hasta 10 metros (desde el inicio). Desarrolle un algoritmo eficiente que resuelva este asunto.

Solución 1: Fuerza Bruta

Una primera solución es generar todas las posibles combinaciones de seleccionar los postes que serán encendidos y quedarnos con aquella que alumbre toda la calle, utilizando la menor cantidad de bombillos. Para ello utilizamos una función generadora de cadenas binarias de tamaño *cantidad de postes*, donde cada 1 de la cadena representa encender el poste de dicha posición.

Correctitud

Interpretaremos una calle de n metros de longitud como un arreglo de n elementos. De esta forma consideramos que colocar un bombillo en la posición i es alumbrar los *2m* metros adyacentes a i, es decir, marcar con 1 el intervalo $[i - m, i + m]$. Entonces una combinación satisface encender la calle completamente si cada posición del arreglo (cada metro de la misma) contiene un 1 luego de encender todos los bombillos de la combinación.

En un primer paso descartamos el hecho de que aún colocando bombillos en todos los postes, no sea posible iluminar la calle completa. Esto se logra comprobando el siguiente lema:

Lema 1

Una calle de longitud n, con postes ubicados en las posiciones $p_1, p_2, \ldots, p_k$ y cumpliendo que $1 \leq p_1 < p_2 < \ldots < p_k \leq n$, es alumbrable con bombillos de alcance m si y solo si:

- El primer poste aparece a una distancia menor o igual que m metros del inicio de la calle: $p_1 - m \leq 1$
- El último poste aparece a una distancia menor o igual que m metros del final de la calle: $p_k + m \geq n$

- Entre cada par de postes hay a lo sumo *2m* metros de distancia, es decir $\forall i, j$ tal que $i < j$: $p_j - p_i \leq 2m + 1$

Cada uno de los subconjuntos de la distribución, si cumple el *Lema 1*, constituye un alumbramiento correcto. La solución será la menor cantidad de bombillos que se requieran para generar alguno de ellos.

Complejidad Temporal

Sea una instancia del problema, donde se tiene como entrada una calle de longitud n, los postes $p_1, p_2, \ldots, p_k$ tales que $1 \leq p_1 < p_2 < \ldots < p_k \leq n$ y bombillos con rango m. Se sigue el siguiente algoritmo:

1. Comprobar que la calle sea alumbrable con la distribución inicial de los postes, es decir, que cumpla con el *Lema 1.* Esto se comprueba con el método `check(n,postes,m)` de complejidad temporal $O(|postes|)$.

2. Si se cumple 1: sea `best = |postes|`, que es la mayor cantidad de postes que pueden ser utilizados y es una posible respuesta.

3. Sea A el conjunto potencia de `postes`. Comprobar para todo conjunto $c \in A$ el cumplimiento del *Lema 1* con el método `check(n,c,m)`; si lo cumple, actualizar `best` como `best = Min(best,t)` donde `t` es la cantidad de 1 que hay en c. Complejidad temporal: $O(|postes| \cdot 2^{|postes|})$

4. Devolver `best` como respuesta.

La complejidad temporal de este algoritmo es:

$$O(|postes|) + O(|postes| \cdot 2^{|postes|})$$

Que por el principio de la suma es:

$$O(|postes| \cdot 2^{|postes|})$$

Solución 2: Dinámica sin optimizar

Toda instancia del problema, donde hay una calle de n metros, un conjunto de postes $p_1, p_2, \ldots, p_k$ tales que $1 \leq p_1 < p_2 < \ldots < p_k \leq n$, con bombillos de alcance m, puede ser interpretado como el siguiente problema T:
Dado un arreglo de tamaño n y un conjunto de intervalos:

$$[p_1 - m, p_1 + m], [p_2 - m, p_2 + m], \ldots, [p_k - m, p_k + m]$$

Determinar cuál es la menor cantidad de intervalos necesaria para cubrir el arreglo de tamaño n.

Lema 2

Existe una biyección entre T y el problema inicial, es decir, una combinación de elementos de $p_1, p_2, \ldots, p_k$ que solucione el problema *Alumbrando Calles* (a partir de ahora AC) es una solución de T, y viceversa.

Demostración Lema 2

Notar que cada intervalo de T coincide con el rango de iluminación de un poste y hay la misma cantidad de postes que intervalos. Por lo que a cada intervalo $[p_i - m, p_i + m]$ de T se le puede asociar el poste p_i en AC, es decir, el rango de iluminación posible de p_i coincide con el intervalo $[p_i - m, p_i + m]$.

Sea S una solución de T, sea P_0 el conjunto de intervalos escogidos para lograr la solución, se cumple que $|P_0| = S$. Escoger todo $p_i \in P_0$ es un alumbramiento correcto de una calle de n metros. Si no es solución de AC es porque existe un conjunto de postes P_0' que es un alumbramiento correcto y $|P_0'| < S$, siendo el conjunto de intervalos asociados a P_0' un cubrimiento completo del arreglo con una cantidad menor que S, lo que no es posible ya que S es solución de T. Entonces se cumple que si S es solución de T implica que también lo es de AC.

Sea S' una solución de AC, sea P_1 el conjunto de postes escogidos para lograr la solución, se cumple que $|P_1| = S'$. Escoger todo $p_i \in P_1$ es una cubrimiento correcto del arreglo con los intervalos $[p_i - m, p_i + m]$. Si no es solución de T es porque existe un conjunto de intervalos P_1' que es un cubrimiento correcto y $|P_1'| < S'$, siendo el conjunto de postes asociados a P_1' un alumbramiento correcto con una cantidad menor que S', lo que no es posible ya que S' es solución de AC. Entonces se cumple que si S' es solución de AC también lo es de T.

La siguiente solución presenta una estrategia dinámica. Sea `dp` un arreglo de tamaño n y que cumple que:
`dp[i]` contiene la menor cantidad de intervalos necesarios para cubrir el arreglo desde la posición 1 hasta la `i`.
En términos del problema *Alumbrando Calles* equivale a decir que:
`dp[i]` contiene la menor cantidad de bombillos necesarios para alumbrar la calle desde la posición 1 hasta la `i`.

Entonces la solución del problema esta en `dp[n]`, ya que sería la cantidad mínima de bombillos necesarios para iluminar el camino desde el inicio hasta n contando con todos los postes disponibles.

Correctitud

Lema 3

En el problema T para todo par de intervalos consecutivos $[p_k - m, p_k + m]$,$[p_{k+1} - m, p_{k+1} + m]$ se cumple que en cada uno hay al menos una posición que no está en el otro.

Demostración Lema 3

Como $p_k < p_{k+1}$ se cumple que $p_k - m < p_{k+1} - m$, por tanto $p_k - m \notin [p_{k+1} - m, p_{k+1} + m]$. Igual se cumple que $p_k + m < p_{k+1} + m$, por lo que $p_{k+1} + m \notin [p_k - m, p_k + m]$.

Lema 4

Sea el conjunto de intervalos $[p_1 - m, p_1 + m], [p_2 - m, p_2 + m], \ldots, [p_k - m, p_k + m]$. Si una posición x no es alcanzable por la izquierda (derecha) del intervalo I_i, es decir, $x < p_i - m$ ($p_i + m < x$), entonces tampoco puede ser alcanzada por ningún intervalo I_j donde $i < j (j < i)$.

Demostración Lema 4

Sea x una posición cualquiera, sea $I_i = [p_i - m, p_i + m]$ el primer intervalo tal que x no es alcanzable por su izquierda (derecha), entonces $x < p_i - m$ ($p_i + m < x$). Aplicando inducción en j para los intervalos I_{i+j} (I_{i-j}):
Si $j = 1$:
Entonces $I_{i+j} = I_{i+1}$ ($I_{i-j} = I_{i-1}$), es el intervalo que está inmediatamente después (antes) de I_i. Por el *Lema 3* en el intervalo I_i va a existir una posición t que no está en I_{i+1} (I_{i-1}), por tanto $t < I_{i+1}$ ($I_{i-1} < t$) y como I_i no alcanza a x por la izquierda (derecha) se cumple que $x < t$ ($t < x$), entonces:

$$x < t < I_{i+1}$$

$$(I_{i-1} < t < x)$$

Entonces x tampoco es alcanzable por la izquierda (derecha) de I_{i+1} (I_{i-1})
Supongamos que se cumple para $j = k$:
Si I_{i+k} (I_{i-k}) lo cumple entonces $x < I_{i+k}$ ($I_{i-k} < x$), aplicando el *Lema 3*, en I_i existe un t que no está en I_{i+k} (I_{i-k}) tal que $t < I_{i+k+1}$ ($I_{i-(k+1)} < t$) y como $x < t$ ($t < x$) entonces:

$$x < t < I_{i+k+1}$$

$$(I_{i-(k+1)} < t < x)$$

Entonces x tampoco es alcanzable por la izquierda (derecha) de I_{i+k+1} ($I_{i-(k+1)}$)
Queda demostrado por inducción que se cumple $\forall j$ con $i < j$ ($j < i$).

Lema 5

En el problema T un intervalo $[p_i - m, p_i + m]$, donde i indica que es el i-ésimo intervalo de los k existentes, puede tener la opción de no utilizarse en el cubrimiento si y solo si se cumple alguno de los siguientes puntos:

1. Si es el primer intervalo se tiene que cumplir que $p_2 - m \leq 1$, es decir que el segundo cubra hasta el inicio del arreglo.

2. Si es el último intervalo se tiene que cumplir que $p_{k-1} + m \geq n$, es decir que el penúltimo cubra hasta el final del arreglo.

3. Si es un intervalo intermedio entonces se tiene que cumplir que $(p_{i+1} - m) - (p_{i-1} + m) \leq 1$, es decir que los intervalos adyacentes a él se solapen o sean continuos.

Demostración Lema 5

Si se cumple alguno de los tres puntos es fácil ver que el intervalo puede no utilizarse, ya que el área que él cubriría en el arreglo puede ser cubierta por otro intervalo.
Demostrando que en caso contrario no puede dejar de utilizarse: sea $[p_i - m, p_i + m]$ un intervalo de los k existentes:

1. Si es el primer intervalo y se cumple que $p_2 - m > 1$, entonces el intervalo $R = [1, p_2 - m - 1]$ no puede ser cubierto por I_2, y por el *Lema 4* tampoco podrá serlo por I_j con $2 < j$. Entonces el único intervalo capaz de cubrir a R es I_1, teniendo que estar presente obligatoriamente en el cubrimiento ya que tiene que cumplirse el *Lema 1*.

2. Si es el último intervalo (I_k) y se cumple que $p_{k-1} + m < n$, entonces el intervalo $R = [p_{k-1} + m + 1, n]$ no puede ser cubierto por I_{k-1}, y por el *Lema 4* tampoco podrá serlo por I_j con $j < k - 1$. Entonces el único intervalo capaz de cubrir a R es I_k, teniendo que estar presente obligatoriamente en el cubrimiento ya que tiene que cumplirse el *Lema 1*.

3. Si es un intervalo intermedio y se cumple que $(p_{i+1} - m) - (p_{i-1} + m) > 1$, hay al menos una posición x tal que $I_{i-1} < x < I_{i+1}$. Entonces x no es alcanzable por la derecha ni la izquierda de I_{i-1} e I_{i+1} respectivamente, por el *Lema 4* se puede afirmar entonces que $\forall j$ donde $j \leq i - 1$ o $i + 1 \leq j$ el intervalo I_j no puede contener a x. Como se cumple *Lema 1*, y todo intervalo distinto de I_i no contiene a x, necesariamente I_i pertenece al cubrimiento.

Queda demostrado por el contrarrecíproco que si el intervalo I_i pude dejar de usarse entonces cumple uno de los tres puntos.

Lema 6

El arreglo `dp` es no decreciente.

Demostración Lema 6

La definición del arreglo plantea que `dp[i]` contiene la menor cantidad de bombillos necesarios para alumbrar la calle desde la posición 1 hasta la `i`. Suponiendo que $\exists i, j$ con $i < j$ tal que $dp[i] > dp[j]$, sea c la combinación de postes que resuelve $d[j]$, por la propia definición de `dp` dicha combinación alumbra desde el inicio hasta j y por tanto también alumbra desde el inicio hasta i. Entonces $dp[i]$ no contiene la menor cantidad de bombillos necesarios para alumbrar la calle desde el inicio hasta i, llegando a una contradicción con la propia definición de `dp`. Por reducción al absurdo se cumple que $\forall i, j$ con $1 \leq i < j \leq n$ se cumple que $dp[i] \leq dp[j]$.

```
1  def solve(n,postes,m):
2      if not check(n,postes,m):
3          return -1
4      dp = [0] * (n + 1)
5      for i in range(0,len(postes)):
6          temp = 1
7          if postes[i] - m - 1 >= 0:
8              temp = dp[postes[i] - m - 1] + 1
9          for j in range(max(1,postes[i] - m),min(postes[i] + m, n) + 1):
10             if dp[j] == 0:
11                 dp[j] = temp
12     return dp[n]
```

En un primer momento, en las líneas 2 y 3 se comprueba que la distribución de postes dada permita alumbrar la calle completamente, a través del método `check(n,postes,m)` verificando el cumplimiento del *Lema 1*.

De la línea 5 a la 11 se realiza un recorrido por todos los postes de izquierda a derecha modificando el arreglo `dp` como se explica a continuación:
Para cada poste p_k se recorre el intervalo $I_k = [p_k - m, p_k + m]$. Por cada posición $i \in I_k$, si $dp[i]$ no está definido (es igual a 0) entonces $dp[i] = dp[p_k - m - 1] + 1$, es decir, incrementar en 1 el valor que hay en la primera posición no alcanzable desde p_k hacia la izquierda, donde $dp[p_k - m - 1] = 0$ si $p_k - m - 1 \leq 1$.

Suponiendo que la entrada dada cumple con el *Lema 1*, demostraremos por inducción que en cada iteración k del algoritmo, el arreglo `dp` se encuentra bien definido desde el inicio hasta la posición $p_k + m$.

k = 1
Como se cumple el *Lema 1*, el intervalo $[1, p_1 + m]$ puede ser alumbrado por un solo bombillo si se pone en el primer poste. Como 1 es la mejor solución posible para poder alumbrar cualquier intervalo se cumple que `dp` está bien definido desde el inicio hasta $p_1 + m$ luego de la primera iteración.

Suponiendo que se cumple $\forall m \leq k$
Sea la iteración $k + 1$ del algoritmo, se cumple que `dp` está bien definido desde el inicio hasta $p_k + m$. Sea $R = [P_k + m + 1, P_{k+1} + m]$ el intervalo de `dp` que debe ser definido en la presente iteración. Se cumple que $\forall i \in R \ dp[i] \geq dp[P_{k+1} - m - 1]$ ya que `dp` es no decreciente por el *Lema 6*.

- ¿Podría $dp[i]$ ser igual a $dp[p_{k+1} - m - 1]$ para alguna $i \in R$?

  Ninguna de las combinaciones de postes que son solución del problema hasta la posición $p_{k+1} - m - 1$ puede contener a p_{k+1}, ya que este no alcanza a ninguna posición del intervalo $[1, p_{k+1} - m - 1]$ y no tiene sentido contarlo. Entonces el poste más cercano a R y que pudiera participar en dichas soluciones es p_k, pero p_k solo alcanza hasta $p_k + m$ y R comienza en $p_k + m + 1$, por lo que R no es alcanzable por la derecha de p_k y por el *Lema 4* tampoco podrá hacerlo ningún poste antes de p_k. Entonces no hay forma de que una combinación que resuelva hasta $p_{k+1} - m - 1$ sea una solución para algún $i \in R$.

- ¿Puede ser $dp[p_{k+1} - m - 1] + 1$ una solución para todo $dp[i]$ con $i \in R$?

  Sea c una combinación cualquiera que resuelve $dp[p_{k+1} - m - 1]$, como $p_{k+1} \notin c$ y $R \subseteq [p_{k+1} - m, p_{k+1} + m]$, se puede afirmar que $c \cup \{p_{k+1}\}$ resuelve $dp[i] \ \forall i \in R$, siendo entonces $dp[i] = dp[p_{k+1} - m - 1] + 1$

Entonces $dp[p_{k+1} - m - 1] + 1$ es la cantidad mínima de postes necesaria para resolver $dp[i] \ \forall i \in R$ y es resuelta en la iteración $k + 1$.

Complejidad Temporal
El algoritmo sigue los siguientes pasos:

1. Comprobar que la distribución de postes dada pueda iluminar toda la calle, igual que en la solución anterior, haciendo uso de `check(n,postes,m)`, con complejidad temporal $O(|postes|)$.

2. Se realizan $|postes|$ iteraciones tal que en la k-ésima iteración `dp[i]` esta correctamente definido $\forall i$ tal que $1 \leq i \leq p_k + m$, basándose en la demostración anterior. Complejidad temporal $O(2m \cdot |postes|)$.

3. Devolver `dp[n]` como resultado del problema.

Entonces su complejidad temporal es

$$O(|postes|) + O(2m \cdot |postes|)$$

Que por el principio de la suma es

$$O(2m \cdot |postes|)$$

.

Solución 3: Dinámica $O(n)$

La siguiente solución está basada en la anterior pero con una pequeña modificación en el código. Como se muestra a continuación:

```
def dinamica_2(n,postes,m):
   if not check(n,postes,m):
      return -1
   last = 0
   dp = [0] * (n + 1)
   for i in range(0,len(postes)):
      temp = 1
      if postes[i] - m - 1 >= 0:
         temp = dp[postes[i] - m - 1] + 1
      for j in range(max(last + 1,postes[i] - m),min(postes[i] + m, n) + 1):
         dp[j] = temp
      last = min(postes[i] + m,n)
   return dp[n]
```

Correctitud
La diferencia con la solución anterior es que, en este caso, en cada iteración k, se puede saber dónde comienza el intervalo R que tiene que ser definido en `dp`, que es `last + 1`. Esto se cumple debido a que en el final de cada iteración i se guarda en `last` el valor $p_i + m$ o n en caso de que llegue al final. (Demostrado en la solución anterior)

Complejidad Temporal
El algoritmo sigue los siguientes pasos:

1. Comprobar que la distribución de postes dada pueda iluminar toda la calle, igual que en la solución anterior, haciendo uso de `check(n,postes,m)`, con complejidad temporal $O(|postes|)$.

2. Se realizan $|postes|$ iteraciones, en cada una de ellas se define en `dp` el intervalo R asociado que no ha sido visitado aún en el algoritmo y una vez que es definido no se vuelve a visitar, por lo que cada posición en `dp` es visitada una única vez. Como $|dp| = n$, la complejidad temporal sería $O(n)$.

3. Devolver `dp[n]` como resultado del problema.

Entonces su complejidad temporal es:

$$O(|postes|) + O(n)$$

Como $|postes|$ es a lo sumo n, por el principio de la suma es:

$$O(n)$$

.

Solución 4: Dinámica $O(|postes|)$

Tomando como experiencia las soluciones 2 y 3, en la iteración $k+1$ del algoritmo, la solución para el intervalo $R = [p_k + m + 1, p_{k+1} + m]$ se basa en la solución para la posición $p_{k+1} - m - 1$. Basado en esto, la presente solución resultó de proponer los siguientes objetivos:

1. En la iteración $k+1$ poder saber el valor de $p_{k+1} - m - 1$.

2. En la iteración k no tener que definir en `dp` las posiciones del intervalo R asociado, sino actualizar la información almacenada de forma tal que la iteración $k+1$ cumpla con el objetivo 1.

3. Si cada una de las iteraciones cumple con el objetivo 2, al final de las $|postes|$ iteraciones se puede saber cual sería la solución para la posición n.

Lema 7

Para toda posición i de `dp`, se cumple que $dp[i-1] = dp[i]$ ó $dp[i] = dp[i+1]$ y en caso de que no suceda es porque en ninguna otra posición j con $j \neq i$ se cumple que $dp[j] = dp[i]$. Es decir, todas las posiciones con igual valor en `dp` están en un único intervalo donde todos los valores son iguales.

Demostración Lema 7

Suponiendo que existen i, j, k tal que $dp[i] = dp[j] \neq dp[k]$, con $i < k < j$, pero esta distribución provocaría el incumplimiento del *Lema 6*.

Lema 8

Sea p_i la posición del i-ésimo poste, entonces en el intervalo $[p_i - m, p_i + m]$ solo puede existir uno o dos valores distintos en `dp`, donde m es la intensidad de los bombillos. En caso de que existan dos valores distintos a, b, con $a < b$, se tiene que cumplir que $a + 1 = b$.

Demostración Lema 8

Suponiendo que existe un poste p_i tal que en el intervalo $I_i = [p_i - m, p_i + m]$ existen más de dos valores distintos. Existen entonces dos posiciones $j, k \in I_i$ tales que $dp[j] < dp[k] - 1$, donde, por el *Lema 6*, $j < k$.

Sea S el conjunto de todas las combinaciones de postes que resuelven $dp[j]$, se cumple por la definición de `dp` que $\forall c \in S$ $|c| = dp[j]$, contemplamos los siguientes casos:

- Si existe un $c \in S$ que use el poste p_i entonces c alumbrará a los intervalos $[1, j]$ e I_i, como $j \in I_i$ ambos intervalos se interceptan, por lo que la combinación c alumbrará el intervalo $[1, j] \cup I_i = [1, p_i + m]$, por tanto a $[1, k]$ también, existe entonces una combinación de cardinalidad $dp[j]$ que resuelve $[1, k]$ con menos de $dp[k]$ bombillos, entrando en contradicción con la definición de `dp` ya que $dp[k]$ no sería la mejor solución.

- Si no existe un $c \in S$ que use el poste p_i entonces toda combinación en S más el poste p_i alumbraría el intervalo $[1, p_i + m]$ y por tanto a $[1, k]$. Entonces $[1, k]$ puede ser resuelto con $dp[j] + 1$ bombillos, llegando a una contradicción con la definición de `dp` ya que $dp[j] + 1 < dp[k]$ y $dp[k]$ no sería la mejor solución.

Queda demostrado entonces por reducción a lo absurdo que se cumple el *Lema 8*.

```
1   def dinamica_3(n,postes,m):
2      if not check(n,postes,m):
3         return -1
4      A = [0]*(len(postes) + 1)
5      A[1] = min(n,postes[0] + m)
6      current = 1
7      if A[1] == n:
8         return 1
9      for i in range(1,len(postes)):
10        if A[max(0,current - 1)] < postes[i] - m - 1:
11           current += 1
12        A[current] = min(n,postes[i] + m)
13        if A[current] == n:
14           return current
15     return -1
```

Correctitud

La idea del algoritmo es la siguiente:

1. Comprobar el cumplimiento del *Lema 1* en las líneas `2-3`.
2. En caso de que se cumpla 1 en las líneas `4-8` se crea el arreglo `A` que en la posición i contiene cuál es la mayor distancia desde el inicio de la calle que puede ser iluminada con i bombillos como mínimo. Se inicializa con el primer bombillo y se comprueba que sea solución. Además se declara la variable `current` que indica la cantidad de bombillos utilizados hasta el momento.
3. En las líneas `9-14` hay un ciclo en el cual, en cada iteración $k+1$, con $k = 0 \dots |postes| - 1$, `current` va a ser igual a la cantidad mínima de postes para resolver el intervalo $R = [p_k + m + 1, p_{k+1} + m]$ asociado a la iteración y actualizando A como `A[current] = postes[k+1] + m`, además la primera vez que se cumpla que `A[current] = n` se devuelve `current`.

En esta solución, se puede saber a través del arreglo `A`, cuáles serían los valores de `dp` en las soluciones 1 y 2 como se muestra a continuación:

$$dp[i] = t \Leftrightarrow A[t-1] < i \leq A[t] \tag{1}$$

Por el *Lema 6* se cumple que `dp` es no decreciente, y en *Lema 8* se afirma que cada intervalo asociado a un poste puede aumentar en 1 como máximo la cantidad de postes de la solución, por lo que se puede afirmar que toda posición i del arreglo `dp` cumple que que $dp[i] = dp[i-1]$ o $dp[i] = dp[i-1] + 1$.

Si se desea saber la cantidad mínima de bombillos necesaria para poder alumbrar la calle desde el inicio hasta la posición $p_{k+1} - m - 1$, se puede lograr aplicando la desigualdad (1), es decir:

$$dp[p_{k+1} - m - 1] = t \Leftrightarrow A[t-1] < p_{k+1} - m - 1 \leq A[t]$$

Entonces se puede afirmar que en cada iteración $k+1$ del algoritmo, con $k = 0 \dots |postes|$, $p_{k+1} - m - 1$ cumple una de las siguientes opciones:

$$\texttt{A[current - 1]} < p_{k+1} - m - 1 \leq \texttt{A[current]}$$

$$\texttt{A[current - 2]} < p_{k+1} - m - 1 \leq \texttt{A[current-1]}$$

Complejidad Temporal

Basándose en la idea del algoritmo se puede analizar su complejidad temporal según los siguientes pasos:

1. En las líneas `2-3` se ejecuta el método `check(n,postes,m)` que es $O(|postes|)$
2. En las líneas `4-8` todas las operaciones son constantes respecto a la entrada
3. Por último está el ciclo de la línea `9`, que se ejecuta $|postes|$ veces y todas las operaciones contenidas son constantes respecto a la entrada.

Entonces la complejidad del algoritmo sería $O(|postes|) + O(|postes|)$, que por el principio de la suma es $O(|postes|)$.

Cota Mínima

Demostraremos mediante la técnica del adversario que no es posible saber si una distribución de postes y de bombillos sobre ellos puede alumbrar la calle completamente haciendo menos preguntas que la cantidad mínima de bombillos necesarios para alumbrarla, con preguntas al estilo ¿hay un bombillo en el poste p_i? Para esto usaremos una calle de $(2m+1) \cdot k$ metros, bombillos de alcance m y un conjunto de k postes $p_1, p_2, \ldots, p_k$ distribuidos como se muestra en la siguiente combinación D:

$$1 \ldots p_1 \ldots 2m+1 \quad (2m+1)+1 \ldots p_2 \ldots 2(2m+1) \quad \cdots \quad (k-1)(2m+1)+1 \ldots p_k \ldots k(2m+1)$$

La distribución D pone los postes de forma tal que sus rangos no se solapen pero que no quede ningún espacio libre. En este caso la cantidad mínima de bombillos necesaria es k, uno por cada poste.

Sea ADV un adversario que dice dar solución al problema sin revisar un bombillo como mínimo. Si se le da a ADV la distribución D existen dos casos posibles:

- Si ADV dice que no es un alumbramiento correcto, se equivoca, ya que D alumbra la calle completamente.
- Si ADV dice que sí es un alumbramiento correcto, sea b_i el bombillo que no revisó, si se le da la distribución D' (D sin b_i), no variaría su respuesta y sería incorrecta ya que el intervalo $[p_i - m, p_i + m]$ solo era alumbrado por el bombillo b_i.