# Informe
# Matemática Discreta II
# Buy a Ticket
# Problema:938D

Abel Molina Sánchez
Grupo 2-11
Ciencias de la Computación
Universidad de La Habana

18 de noviembre de 2020

# 1. Problema:

D. Buy a Ticket

Musicians of a popular band 'Flayer' have announced that they are going to 'make their exit' with a world tour. Of course, they will visit Berland as well.
There are $n$ cities in Berland. People can travel between cities using two-directional train routes; there are exactly $m$ routes, i-th route can be used to go from city $v_i$ to city $u - i$ (and from $u_i tov_i$), and it costs $w_i$ coins to use this route.
Each city will be visited by 'Flayer', and the cost of the concert ticket in i-th city is $a_i$ coins.
You have friends in every city of Berland, and they, knowing about your programming skills, asked you to calculate the minimum possible number of coins they have to pay to visit the concert. For every city i you have to compute the minimum number of coins a person from city $i$ has to spend to travel to some city $j$ (or possibly stay in city i), attend a concert there, and return to city $i$ (if $j \neq i$).
Formally, for every you have to calculate , where $d(i,j)$ is the minimum number of coins you have to spend to travel from city $i$ to city $j$. If there is no way to reach city $j$ from city $i$, then we consider $d(i,?j)$ to be infinitely large.

**Input:** The first line contains two integers $n$ and $m$ ($2 \leq n \leq 2 * 10^5$, $1 \leq m \leq 2 * 10^5$).
Then $m$ lines follow, i-th contains three integers $v_i$, $u_i$ and $w_i(1 \leq v_i, u_i \neq u_i, 1 \leq w_i \leq 10^{12})$ denoting i-th train route. There are no multiple train routes connecting the same pair of cities, that is, for each $(v,u)$ neither extra $(v,u)$ nor $(u,v)$ present in input.
The next line contains $n$ integers $a_1, a_2, ..., a_k$ ($1 \leq a_i \leq 10^{12}$) — price to attend the concert in i-th city.

**Output:** Print $n$ integers. $i$-th of them must be equal to the minimum number of coins a person from city i has to spend to travel to some city $j$ (or possibly stay in city $i$), attend a concert there, and return to city $i$ (if $j \neq i$).

Interpretación del problema:

Tenemos un grafo $G = (V,E)$ no dirigido y ponderado, y se pide optimizar $\forall$ ciudad $c_i$ el costo para asistir al concierto. Se le agrega al grafo la dificultad de de que los vértices también poseen un costo, que sería el pago del concierto en la ciudad, luego el problema no es por vía directa buscar el camino de costo mínimo para cada cada vértice del grafo, aunque desde un principio se tiene que los caminos de costo mínimo forman parte de la solución.
Luego la idea intuitiva siempre fue intentar reducir el problema a un problema de cálculo de caminos de costo mínimo, para el cual conozco el algoritmo de Djikstra.
La estrategia de solución presentada consiste en agregar un nuevo vértice al grafo, llamemosle f, de tal forma que este vértice tiene aristas con todos los vértices del grafo, dandole a la arista $(f,i)$ que conecta a f con la ciudad i, el costo de ver la película en esta ciuadad, luego hay buscar los camninos de costo mínimo desde este vértice hacia todos los del grafo utiliznado el algoritmo de Djikstra. Para esto, hay que demostrar que el camino de costo mínimo de f a i es el camino de costo mínimo de i hasta la ciudad j(posible j = i) siendo j la opción óptima para i.

## 2. Demostración:

**Definición:** Sea un grafo $G = (V, E)$ se dice que G es ponderado si existe una función de costo $w \rightarrow N_+$[1] tal que a cada arista $(u, v) \in E, w(uv)$ es el costo de esa arista

**Definición:** Sea $G = (V, E)$ un grafo ponderado y sea p un camino entre u y v, $p = u = v_0, v_1, ...v_k$, el costo de p es la suma de los costos de las aristas que unen los vértices que forman parte de p.
$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$

**Definición:** Sea $G = (V, E)$ un grafo ponderado, se dice que p es un camino de costo mínimo de u a v, si ($w(p)$ es igual al mínimo entre el costo de todos los caminos que unen u y v.
$w(p) = min(\{w(q) : \forall q \text{ camino de } u \rightarrow v\})$

**Notación:** Se denota como $\delta(u, v)$ al costo de los caminos de costos mínimos de u a v.

**Definición:** Sea $G = (V, E)$ no dirigido y ponderado, $u, v \in V$.
Sea p un camino de u a v, tal que $p = \{u = v_0, v_1, v_{k-1}, v_k = v\}$ en $G$, $p^-$ será el camino de $v$ a $u$ tal que $p^-1 = \{v = v_k, v_{k-1}, v_{k-2}, ...v_1, v_0 = u\}$ tal que contiene las mismas aristas de p pero en orden contrario.

**Propiedad 1**: $w(p) = w(p^-)$

*Demostración*: Directo de la definición de $p^-$, si $w(p) \neq w(p^-)$ tendría que existir un p' o en p una arista que no esté en p o en $p^-$, lo cual es una contradicción.

***Lema 1**: Sea $G(V, E)$ un grafo no dirigido y ponderado y sea p un camino de costo mínimo de $u \rightarrow v$, $p^-$ es también un camino y de costo mínimo de $v \rightarrow u$ y $\delta(u, v) = \delta(v, u)$.*

*Demostración:* Supongo que $p^-$ no es el camino de costo mínimo de $v \rightarrow u$, entonces existe un camino q de $v \rightarrow u$ tal que $w(q) < w(p^-)$. Luego como el grafo es no dirigido, existe $q^-$ que es también un camino de u a v cuyo costo es menor que p, pero esto es una contradicción con el hecho de que p sea un camino de costo mínimo de $u \rightarrow v$. Por tanto $p^-$ es un camino de costo mínimo de $u \rightarrow v$ y $w(p^-) = \delta(v, u)$. Por tanto como $w(p) = w(p^-)$, $\delta(u, v) = \delta(v, u)$

***Lema 2:** Los caminos de costo mínimo en un grafo $G = (V, E)$ ponderado son caminos simples.*

*Demostración:* Supongamos que existe un camino de costo mínimo que no es simple. Llamémosle p. Como p no es simple, entonces p contiene ciclos.
Eliminando los ciclos de p, se obtuvo un camino p' tal que

$w(p') = w(p) - w(C_i)$, siendo $C_i$) las aristas que formaban ciclos en p $w(C_i) > 0$, luego

[1] se define sobre la restricción del ejercicio, en sentido general $w = E \in R$.

$w(p') < w(p)$ lo cual es una contradicción con el hecho de que p es un camino de costo mínimo. Luego los caminos de costo mínimo son simples.

***Lema 3:*** *Sea $G = (V, E)$ un grafo no dirigido y ponderado, el costo mínimo de ir de u a v y de v a u es $2\delta(u, v)$.*

*Demostración:*
Sea p un camino de costo mínimo de u a v y sea que un camino de costo mínimo de v a u (posible $q = p^{-}$). El costo mínimo de ir de u a v y de v a u
$w(uv; vu) = w() + w(q)$
$\delta(u, v) + \delta(v, u)$ y por Lema 1 $\delta(u, v) = \delta(v, u)$
$\delta(u, v) + \delta(u, v)$
$2\delta(u, v)$

**Notación:** Denotemos $\tau(v)$ al costo mínimo de ir desde u, v y regresar de v a u.

***Lema 4:*** *Sea el grafo $G = (V_G, E_G)$ no dirigido y ponderado con función de costo $w_E = E_G \rightarrow N_+$, si se define el grafo $H = (V_H = V_G = E_H = E_G)$ no dirigido y ponderado cuya función de costo $w_H = E_H \rightarrow N_+$ se difine como $w_H(u, v) = 2w_G(u, v)$, $\forall (u, v) \in E_H$ se cumple que $\delta(u, v)$ en $H$ es igual $\tau(u, v)$ en $G$.*

*Demostración:* Por definición $H$ y $G$ contienen los mismos vértices y las mismas aristas, difieren en el costo, siendo el costo de cada arista en $H$ el doble del costo de la arista en G. Luego tenemos $p_G u = v_0, v_1, ..., v_k = v$ un camino que une a u y v en G, digamos que $p_H$ es el camino que une a u y v en $H$, cuyos vértices y aristas son los mismos, difiriendo entre uno y otro en el costo. Entonces, por definición de $w_H$

$w(p_H) = \sum_{i=1}^{k} 2w_G(v_{i-1}, v_i)$
$w(p_H) = 2\sum_{i=1}^{k} w_G(v_{i-1}, v_i)$, $\sum_{i=1}^{k} w_G(v_{i-1}, v_i) = w(p_G)$
$w(p_H) = 2w(p_G)$

luego $forall$ $p_G$ camino en $G$ se cumple que $w(p_H) = 2w(p_G)$ y por tanto también se cumplirá en los caminos de costo mínimo, por lo cual
$\delta_H(u, v) = 2\delta_G(u, v)$ y como $\tau(u, v) = 2\delta(u, v)$
$\delta_H(u, v) = \tau_G(u, v)$.

***Lema 5:*** *Sea $G = V, E)$ un grafo no dirigido y ponderante y sea p un camino de costo mínimo de u a v, $p = u = v_0, v_1, v_2, ...v_{k-1}, v_k = v$, $p_i = u = v_0, ...v_i$ es un camino de costo minimo de u a $v_i = \forall i = 1, 2, ...k - 1$.*

*Demostración:* Supongamos que $\exists i$ tal que $p_i$ no es un camino de costo mínimo de u a $v_i$, entonces existe un camino $f_i$ tal que $q_i$ es un camino de costo mínimo de u a $v_i$ y $w(q_i) < w(p_i)$. Luego si se forma el camino t de u a v, tal que $t = q_i, v_{i+1}, ...v_k = v$ $w(t) < w(p)$, lo cual es una contradicción con el hecho de que p es un camino de costo mínimo de u a v. Luego el Lema 5 se cumple $v_i$).

***Teorema:*** *Dado el problema inicial, la solución planteada es correcta.*

*Demostración:*
Sea $G = (V_G, F_G)$ el grafo de entrada y sea $a_i$ el costo de asistir al concierto en la ciudad, $\forall$ $i$,$i = 1, ...|V_G|$
Sea $H = (V_H = V_G, E_H = E_G)$ un grafo no dirigido y ponderado cuya función de costo $w_H = E_H \to N_+$, se define como $w_H(u, v) = 2w_E(u, v)$ $\forall(u, v) \in E_H$, por el Lema 4 se tiene que $\delta_H(u, v) = \tau_G(u, v)$.

Sea ahora un vértice f, se agrega f a $H$, tal que $H \cup f = (V_{H\cup f}, E_H \cup \{(f, v)\forall v \in V_H\}$ y $w(f, v_i) = a_i$ $\forall i, 1 = 1, ...|V_G|$

Esto es, se agrega a $H$ un vértice tal que posee aristas que lo conectan con cada vértice del grafo, y el costo de estas aristas será igual al costo de asistir al concierto en la ciudad $i$ en el problema inicial.

Luego $\forall v \in V_G = V_H = V_{H-f}$ se cumple que existe un camino de $v$ a $f$, ya que existen las aristas $(f, v_i)$.

Sea
$p_i = v_i = v_0, v_1, ...v_k, v_{k+1} = f$, el camino de costo mínimo desde $v_i$ hasta f.

$p_i = p_{ik} = \{v_i = v_0, ...v_k\} \cup v_k, f)$ y $w(p_i) = w(p_{ik})_{H\cup f} + w(v_k, f)_{H\cup f}$ y por Lema 5 se tiene que $p_{ik}$ es un camino de costo mínimo desde $v_i$ a $v_k$ y $w(v_k, f) = a_k$ y por Lema 3 $w(p_{ik})_{H\cup f} = 2w(p_{ik})_G$ y $w(v_k, f) = a_k$.

Supongamos que $p_i$ no es la mejor opción para la ciudad, entonces existe una ciudad $j$, $j \neq i$, tal que 2 veces el costo de ir de i a j, más el costo de asistir al concierto en la ciudad j es menor que ir y asistir al concierto en la ciudad k, pore esto es que existiría un camino $q^j_G$ en $G$ de $v_i$ a $v_j$ y que
$2w(q^j_G) + a_j < w(p_i)_{H\cup f}$

Pero esto sería que existe un camino $q^i j_H)$ tal que
$w(q^j_H) + a_j < w(p_i)_{H\cup f}$,

Pero como $q^i_H$ es un camino en $H \cup f$ y $a_j = w(v_j, f)_{H\cup f}$, entonces existe un camino
$q_i = q^j_H \cup \{(v_i, f)\}$ que une a $v_i$ con $f$ en $H \cup f$ tal que
$w(q_i)_{H\cup f} < w(p_i)_{H\cup f}$,
lo cual es una contradicción con el hecho de que $p_i$ era el camino de costo mínimo.

Cuando la mejor opción para la ciudad i es asistir al concierto en la misma ciudad $i$, se demuestra análogamente, siendo un caso específico donde $p_i = (v_i, f)$ y $w(p_i) = w(v_i, f) = a_i$

Luego hemos demostrado llevando el problema de su condición A a una condición C que la solución en C es solución en A.

Ahora, como por Lema 1 se tiene que dado que sea $p$ un camino de costo mínimo, $p^-$ será también de costo mínimo y $\delta(v,u) = \delta(v,u)$ como $p_i$ de costo mínimo de $v_i af$ es solución del problema, lo será también $p_i^-$ de $f$ a $v_i$ por dicho Lema 3.

Utilizando este resultado la solución se implementa corriendo el algoritmo de Dijkstra desde el vértice f sobre el grafo $H \cup f$ para obtener los caminos de costo mínimo desde f, hasta $v_i \forall i, 1 = 1, ...|V|$, que por todo lo demostrado anteriormente constituye una solución correcta al problema inicial.
La correctitud del algoritmo de Dijkstra, se pasa a demostrar a continuación.

```
 Djikstra(G,s):
1    por cada vértice v en V[G]:
2         d[v] = INF
3    d[s] = 0, Q=V
4    mientras haya vértices en Q:
5         extraer u de Q tal que d[u] es mínima en Q
6         Q = Q-{u}
7         por cada vértice v en Q adyacente a u
8             d[v] = min(d[v],d[u]+w(u,v))
```

***Teorema:*** *Luego de la ejecución del algoritmo de Dijkstra* $(G,s)$
$d[v] = \in (s,v) \forall v \in V[G]$.

*Demostración:*
Como $\delta(s,v)$ es el mínimo de s a v, se tiene que: $(1)\delta(s,v) \leq d([v]$ siendo $d[v]$ el costo del camino de s a $v$, mediante el cual es alcanzado $v$ durante la ejecución del algoritmo.
Busquemos la igualdad demostrando que $d[v] \leq \delta(s,v)$. Inducción sobre el orden es que son extraídos los vértices de Q.
Caso base: el primer nodo extraído será S, y se tiene que $d[s] = 0 = \delta(s,s)$, lo cual se cumple.

Ahora supongamos por hipótesis que se cumple para todo vértice v que fue extraído de Q antes que un vértice u.
Ahora tengamos $s = v_0, v_1, ...v_{a-1}, v_n = u$ un camino de costo mínimo de s a u, luego por Lema 5,
$\delta(s,v) = \sum_{k-1}^{j} w(v_{j-1}, v_j)$, $\forall j, j = 0, ...n$

Sea $i$ el máximo, tal que $v_i$ fue extraído de Q antes que u, por hipótesis se tiene que $\delta(s,v) > d[v_i]$ y como por (1) se tiene $\delta(s,v_i) \leq d[v_i]$ entonces
$\delta(s,v_i) - d[v_i] = \sum_{j-1}^{i} w(v_{j-1}, v_j)$.

Ahora, durante la iteración en la cual $v_i$ fue extraído de Q se tiene que para $v_{i+1}$ adyacente a $v_i$, $d[v_{i+1}] \leq d[v_i] + w(v_i, v_{i+1})$
luego de terminada la iteración, lo cual se garantiza por la línea 8. Luego, como $d[v_i+1]$ no aumenta en ningún momento del algoritmo, se seguirá cumpliendo que $d[v_i+1]\ leqd[v_i] + w(v_i, v_{i+1})$ en la iteración en la que u es extraído de Q, luego se tiene

(3) $d[v_{i+1} \leq d[v_i] + w(v_i, v_{i+1} = \delta(s, v_i) + w(v_i, v_{i+1} = \delta(s, v_{i+1} = \delta(s, u)$.

Supongamos que $v_{i+1} \neq u$, esto implica que $i \neq n-1$, luego antes de extraerse u, en una iteración anterior se tendría $\delta[s,u]vi+1 < d[u]$ por el camino de costo mínimo, lo cual implica (por 3) que $d[v_{i+1} < d[v]$, pero eso es una contradicción con el hecho de que i era el máximo tal que $v_i$ se extrajo primero que u de Q. Luego tiene que $v_{i+1} = u$, con lo cual se cumple $d[u] \leq \delta(s,u)$ lo cual implica que $d[u] = \delta(s,u)$. Luego de terminada la ejecución del algoritmo de Dijkstra $d[v] = \delta(s,v) \forall verticev \in V[G]$.

El algoritmo de Djikstra implementedo con cola con prioridad tiene un complejidad de temporal de $O(|E|log|V|)$, ya que por cada vértice se recorre su lista de adyacencia, lo que, luego de finalizado el algoritmo se ha recorrió toda la lista de adyacencia del grafo. Cada operación de extraer el mínimo desde se realiza es $O(log|V|)$ y se realiza $|V|$ veces, se tiene , cada operación para actualizar en la cola de prioridad es $O(logV)$ y se realiza a lo sumo por cada arista, o sea $O(|E|)$, por lo tanto se tiene que se realiza $O(|V| + |E|)$ operaciones sobre la cola con prioridad, por lo cual la complejidad del algoritmo es $O(|V|log|V| + |E|log|V|)$ que es $O(|E|log|V|)$. Crear el heap es $O(log|V|)$. Se utiliza la estructura del módulo heap q de python.