

ANÁLISIS DEL RETO 3

Juliana Rodríguez Morales – 202421552 – js.rodriguezm1234

Maria Clara Quijano - 202420069 - m.quijanoa

Juan Andrés Lozada - 202510410-j.lozadab

Requerimiento 1

Descripción

Dada una aerolínea y un rango de minutos quiero identificar los vuelos que presentan un retraso en la hora de salida que están dentro del rango.

```
def req_1(catalog, aerolinea, rango):
    """
    ...
    start = get_time()

    rango[0] = int(rango[0])
    rango[1] = int(rango[1])
    trayectos = 0
    viajes_filtrados = pq.new_heap(True)
    viajes = catalog["aerolinea"] #Mapa
    aero = mp.get(viajes, aerolinea) #Me da un mapa de con llaves de aeropuertos
    codigos = mp.key_set(aero)
    for i in range(sl.size(codigos)):
        lista = mp.get(aero, lt.get_element(codigos, i))
        for j in range(sl.size(lista)):
            elem = sl.get_element(lista, j)
            retraso = diferencia_tiempo(elem["dep_time"], elem["sched_dep_time"])
            if rango[0] <= retraso and rango[1] >= retraso:
                trayectos += 1
                viaje = {"Id": elem["id"],
                         "Fecha": elem["date"],
                         "Nombre Aerolínea + Código": elem["name"] + " - " + elem["carrier"],
                         "Origen": elem["origin"],
                         "Destino": elem["dest"],
                         "Retraso": retraso
                }
                pq.insert(viajes_filtrados, (retraso, dt.datetime.strptime(elem["date"]+" "+elem["dep_time"], "%Y-%m-%d %H:%M"), viaje))
    end = get_time()
    tiempo = delta_time(start, end)
    return tiempo, trayectos, viajes_filtrados
```

Entrada

- Código de la aerolínea a analizar.

- Rango de minutos de retraso en salida a filtrar.

Salidas

Tiempo de la ejecución del requerimiento en milisegundos. Número total de vuelos que cumplen con el filtro de la aerolínea y rango de retraso. Teniendo en cuenta los vuelos que cumplen el filtro, presente los siguientes datos:

- ID del vuelo.
- Código del vuelo.
- Fecha.
- Nombre de la aerolínea.
- Código de la aerolínea.
- Aeropuerto de origen.
- Aeropuerto de destino.
- Minutos de retraso en la salida.

Implementado (Sí/No)

Sí, por Juliana Rodríguez

Análisis de complejidad

Pasos

Pasos	Complejidad
Obtener elemento de un mapa (submapa)	$O(n/N)$
Crear un single con las llaves del mapa	$O(n)$
Primer for (Recorre todos los submapas)	$O(n)$
Obtener elemento del submapa (lista)	$O(m/M)$
Segundo for (Recorre los trayectos de la lista)	$O(m)$
Obtener elemento de la lista (viaje)	$O(m)$
Insertar en una Cola de Prioridad	$O(1+\log m)$
TOTAL	$O(n*m(1+\log m))$

Análisis

Dado que la estructura con la información necesaria para este requerimiento era un mapas (aeropuertos) dentro de un mapa (de aerolíneas) se utilizaron dos recorridos for para poder obtener la información de todos viajes filtrados. Asimismo, la complejidad es afectada al meter los datos dentro de una cola de prioridad aquellos vuelos que tenían un retraso dentro del rango. Aunque son dos recorridos estos tienen complejidad de variables diferentes ya que el primero recorre todos los mapas dentro de la aerolínea filtrada, y el segundo recorre todos los viajes dentro del mapa previamente recorrido, lo que encadena la complejidad.

Pruebas (Al menos 5)

Hechos con large:

Diga el código de la aerolínea: UA					
Diga el mínimo del rango de minutos (a): 10					
Diga el máximo del rango de minutos (b): 30					
Tiempo de ejecución: 7856.79					
Número de viajes filtrados: 6664					
Primeros viajes filtrados:					
Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
231968	2013-01-01	United Air Lines Inc. - UA	EWR	PHX	10
10247	2013-01-02	United Air Lines Inc. - UA	EWR	ORD	10
190900	2013-01-02	United Air Lines Inc. - UA	EWR	MCO	10
225922	2013-01-03	United Air Lines Inc. - UA	EWR	BOS	10
45089	2013-01-05	United Air Lines Inc. - UA	EWR	DEN	10
Últimos viajes filtrados:					
Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
177540	2013-12-26	United Air Lines Inc. - UA	EWR	FLL	30
116068	2013-12-26	United Air Lines Inc. - UA	EWR	IAH	30
207125	2013-12-26	United Air Lines Inc. - UA	EWR	IAH	30
101930	2013-12-27	United Air Lines Inc. - UA	EWR	FLL	30
159718	2013-12-28	United Air Lines Inc. - UA	EWR	PDX	30

Diga el código de la aerolínea: AA
 Diga el mínimo del rango de minutos (a): 1
 Diga el máximo del rango de minutos (b): 35
 Tiempo de ejecución: 4717.96
 Número de viajes filtrados: 6175
 Primeros viajes filtrados:

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
212429	2013-01-01	American Airlines Inc. - AA	LGA	DFW	1
23327	2013-01-02	American Airlines Inc. - AA	JFK	LAS	1
265141	2013-01-03	American Airlines Inc. - AA	EWR	DFW	1
181338	2013-01-03	American Airlines Inc. - AA	JFK	LAX	1
137835	2013-01-04	American Airlines Inc. - AA	JFK	SFO	1

Últimos viajes filtrados:

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
137931	2013-12-15	American Airlines Inc. - AA	LGA	ORD	35
128953	2013-12-18	American Airlines Inc. - AA	LGA	ORD	35
66005	2013-12-22	American Airlines Inc. - AA	JFK	MIA	35
177108	2013-12-26	American Airlines Inc. - AA	LGA	MIA	35
79530	2013-12-28	American Airlines Inc. - AA	LGA	DFW	35

Diga el código de la aerolínea: FL
 Diga el mínimo del rango de minutos (a): 40
 Diga el máximo del rango de minutos (b): 60
 Tiempo de ejecución: 145.45
 Número de viajes filtrados: 123
 Primeros viajes filtrados:

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
27634	2013-03-18	AirTran Airways Corporation - FL	LGA	ATL	40
267131	2013-12-04	AirTran Airways Corporation - FL	LGA	ATL	40
243767	2013-03-14	AirTran Airways Corporation - FL	LGA	CAK	41
2013	2013-04-01	AirTran Airways Corporation - FL	LGA	CAK	41
26708	2013-05-21	AirTran Airways Corporation - FL	LGA	ATL	41

Últimos viajes filtrados:

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
129482	2013-07-11	AirTran Airways Corporation - FL	LGA	ATL	59
152392	2013-09-01	AirTran Airways Corporation - FL	LGA	CAK	60
252063	2013-11-03	AirTran Airways Corporation - FL	LGA	ATL	60
265640	2013-11-07	AirTran Airways Corporation - FL	LGA	ATL	60
436	2013-12-22	AirTran Airways Corporation - FL	LGA	ATL	60

Diga el código de la aerolínea: Ha
 Diga el mínimo del rango de minutos (a): 10
 Diga el máximo del rango de minutos (b): 15
 Tiempo de ejecución: 6.05
 Número de viajes filtrados: 4

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
135112	2013-08-21	Hawaiian Airlines Inc. - HA	JFK	HNL	11
289153	2013-05-31	Hawaiian Airlines Inc. - HA	JFK	HNL	13
155769	2013-01-03	Hawaiian Airlines Inc. - HA	JFK	HNL	14
65214	2013-02-18	Hawaiian Airlines Inc. - HA	JFK	HNL	14

Diga el código de la aerolínea: 9e
 Diga el mínimo del rango de minutos (a): 5
 Diga el máximo del rango de minutos (b): 20
 Tiempo de ejecución: 268.78
 Número de viajes filtrados: 1736
 Primeros viajes filtrados:

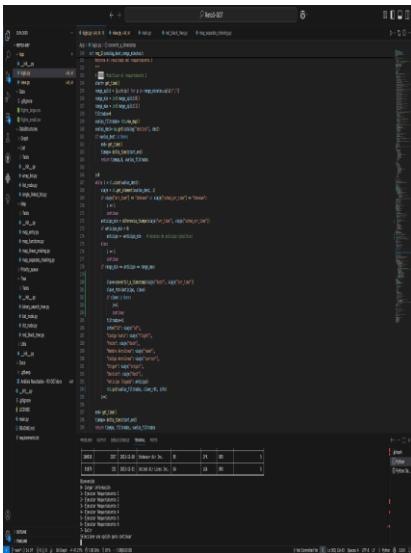
Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
122057	2013-01-01	Endeavor Air Inc. - 9E	JFK	JAX	5
9405	2013-01-03	Endeavor Air Inc. - 9E	JFK	ROC	5
30076	2013-01-04	Endeavor Air Inc. - 9E	JFK	BWI	5
236079	2013-01-05	Endeavor Air Inc. - 9E	JFK	IND	5
204768	2013-01-06	Endeavor Air Inc. - 9E	JFK	PIT	5

Últimos viajes filtrados:

Id	Fecha	Nombre Aerolínea + Código	Origen	Destino	Retraso
106339	2013-12-13	Endeavor Air Inc. - 9E	JFK	BOS	20
113439	2013-12-18	Endeavor Air Inc. - 9E	EWR	DTW	20
278421	2013-12-20	Endeavor Air Inc. - 9E	JFK	PIT	20
122063	2013-12-21	Endeavor Air Inc. - 9E	JFK	RDU	20
274403	2013-12-29	Endeavor Air Inc. - 9E	JFK	IND	20

Promedio de tiempos: 2599.006 ms

Requerimiento 2



The screenshot shows a Java code editor with several tabs open. The main tab contains a class named `FlightArrivalFilter` with a method `filterArrivals`. The code uses a `TreeMap` to store flight arrivals by destination and then iterates through each entry to check if the arrival time falls within a specified range relative to the destination's local time. It also includes logic to handle time zone differences and filter flights based on a specific code.

```
public class FlightArrivalFilter {
    public static void filterArrivals(TreeMap<String, List<FlightArrival>> arrivals, String destination, int minutes) {
        for (Map.Entry<String, List<FlightArrival>> entry : arrivals.entrySet()) {
            String dest = entry.getKey();
            List<FlightArrival> arrivalsForDest = entry.getValue();
            for (FlightArrival arrival : arrivalsForDest) {
                if (arrival.getArrivalTime().getLocalTime().isBefore(dest.getOffsetTime().plusMinutes(minutes))) {
                    arrival.setArrivedEarly(true);
                }
            }
        }
    }
}
```

Descripción

Toman los datos de vuelos y les saca el anticipo en la hora de llegada dentro de un rango de minutos si pasan el filtro del aeropuerto destino.

Entrada

- Código destino

Salidas

- Rango minutos de anticipo en la llegada a filtrar
- Tiempo de ejecución
- Número total de vuelos que cumplen con el filtro del aeropuerto y del rango de anticipo
- Vuelos que cumplen el filtro:
 - o Id
 - o Código
 - o Fecha
 - o Nombre
 - o Aeropuerto origen
 - o Aeropuerto destino
 - o Minutos anticipio en salida

Implementado (Sí/No)

Si, Maria Clara Quijano

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos

Busqueda por destino

Complejidad

$O(\log N)$

Calculo anticipado	$O(1)$
While	$O(m)$
Get y put	$O(\log k)$
TOTAL	$O(m * \log k)$

Análisis

Se hace un filtro por búsqueda, y luego entra al while, lo cual es lo que incrementa más su complejidad ya que itera sobre m vuelos, los que tienen el destino especificado. Normalmente, n será un valor bajo. Por el uso del Red Black Tree, la complejidad de insert y search es de $O(\log k)$, y mantiene un orden ascendente por anticipado de manera eficiente.

Pruebas (Al menos 5)

Hechos con large:

Diga el código del aeropuerto de destino: LAX

Diga el rango de minutos de antícpo (min,max): 10,30

== Requerimiento 2: Vuelos con antícpo en la llegada ==

Tiempo de ejecución (ms): 60062.15

Número de vuelos filtrados: 4477

-- Primeros 5 vuelos filtrados (Menor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
184234	407	2013-01-04	Virgin America	VX	JFK	LAX	10
35106	117	2013-01-05	American Airlines Inc.	AA	JFK	LAX	10
302589	161	2013-01-06	United Air Lines Inc.	UA	JFK	LAX	10
154706	863	2013-01-13	Delta Air Lines Inc.	DL	JFK	LAX	10
66470	181	2013-01-16	American Airlines Inc.	AA	JFK	LAX	10

-- Últimos 5 vuelos filtrados (Mayor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
218175	997	2013-12-13	United Air Lines Inc.	UA	JFK	LAX	30
238075	117	2013-12-15	American Airlines Inc.	AA	JFK	LAX	30
142338	185	2013-12-25	American Airlines Inc.	AA	JFK	LAX	30
108792	785	2013-12-26	United Air Lines Inc.	UA	JFK	LAX	30
288462	477	2013-12-30	Delta Air Lines Inc.	DL	JFK	LAX	30

Diga el código del aeropuerto de destino: ORD

Diga el rango de minutos de antícpo (min,max): 1,5

== Requerimiento 2: Vuelos con antícpo en la llegada ==

Tiempo de ejecución (ms): 63634.58

Número de vuelos filtrados: 1127

-- Primeros 5 vuelos filtrados (Menor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
240360	1200	2013-01-02	United Air Lines Inc.	UA	EWR	ORD	1
124574	3795	2013-01-03	Envoy Air	MQ	EWR	ORD	1
92206	980	2013-01-03	United Air Lines Inc.	UA	EWR	ORD	1
272551	3730	2013-01-03	Envoy Air	MQ	EWR	ORD	1
96011	627	2013-01-06	United Air Lines Inc.	UA	EWR	ORD	1

-- Últimos 5 vuelos filtrados (Mayor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
71530	331	2013-12-27	United Air Lines Inc.	UA	LGA	ORD	5
1169	219	2013-12-28	United Air Lines Inc.	UA	EWR	ORD	5
69158	279	2013-12-28	United Air Lines Inc.	UA	EWR	ORD	5
160918	2927	2013-12-30	Endeavor Air Inc.	9E	JFK	ORD	5
91079	331	2013-12-31	United Air Lines Inc.	UA	LGA	ORD	5

Diga el código del aeropuerto de destino: CLT
Diga el rango de minutos de antícpo (min,max): 20,60

== Requerimiento 2: Vuelos con antícpo en la llegada ==
Tiempo de ejecución (ms): 32591.26
Número de vuelos filtrados: 1634

-- Primeros 5 vuelos filtrados (Menor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
53525	4582	2013-01-05	Envoy Air	MQ	LGA	CLT	20
284845	1447	2013-01-05	US Airways Inc.	US	LGA	CLT	20
85177	675	2013-01-07	US Airways Inc.	US	EWR	CLT	20
102496	375	2013-01-08	US Airways Inc.	US	EWR	CLT	20
60963	720	2013-01-12	US Airways Inc.	US	EWR	CLT	20

-- Últimos 5 vuelos filtrados (Mayor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
238016	5475	2013-10-27	ExpressJet Airlines Inc.	EV	LGA	CLT	45
269882	4714	2013-09-05	ExpressJet Airlines Inc.	EV	EWR	CLT	46
265832	1085	2013-05-14	JetBlue Airways	B6	JFK	CLT	47
200118	4714	2013-09-09	ExpressJet Airlines Inc.	EV	EWR	CLT	49
75814	4978	2013-09-04	ExpressJet Airlines Inc.	EV	LGA	CLT	50

Diga el código del aeropuerto de destino: BWI
Diga el rango de minutos de antícpo (min,max): 5,5

== Requerimiento 2: Vuelos con antícpo en la llegada ==
Tiempo de ejecución (ms): 612.64
Número de vuelos filtrados: 28

-- Primeros 5 vuelos filtrados (Menor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
215969	3635	2013-01-03	Endeavor Air Inc.	9E	JFK	BWI	5
51724	477	2013-01-08	Southwest Airlines Co.	WN	EWR	BWI	5
240085	477	2013-01-30	Southwest Airlines Co.	WN	EWR	BWI	5
197498	477	2013-01-31	Southwest Airlines Co.	WN	EWR	BWI	5
78225	808	2013-02-04	Southwest Airlines Co.	WN	EWR	BWI	5

-- Últimos 5 vuelos filtrados (Mayor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
189952	3885	2013-10-21	Endeavor Air Inc.	9E	JFK	BWI	5
34015	4249	2013-10-25	ExpressJet Airlines Inc.	EV	EWR	BWI	5
182172	2940	2013-11-03	Endeavor Air Inc.	9E	JFK	BWI	5
173745	3044	2013-11-13	Envoy Air	MQ	JFK	BWI	5
40531	4234	2013-12-30	ExpressJet Airlines Inc.	EV	EWR	BWI	5

```
Diga el código del aeropuerto de destino: PHL
Diga el rango de minutos de antícpo (min,max): 1,100

== Requerimiento 2: Vuelos con antícpo en la llegada ==
Tiempo de ejecución (ms): 1381.3
Número de vuelos filtrados: 742
```

-- Primeros 5 vuelos filtrados (Menor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
259019	3271	2013-01-10	ExpressJet Airlines Inc.	EV	EWR	PHL	1
63927	1833	2013-01-17	US Airways Inc.	US	JFK	PHL	1
24069	3689	2013-01-18	Endeavor Air Inc.	9E	JFK	PHL	1
25985	1896	2013-03-02	US Airways Inc.	US	LGA	PHL	1
235731	3650	2013-03-09	Endeavor Air Inc.	9E	JFK	PHL	1

-- Últimos 5 vuelos filtrados (Mayor antícpo, Cronológico) --

Id	Código Vuelo	Fecha	Nombre Aerolínea	Código Aerolínea	Origen	Destino	Antícpo llegada
86969	4218	2013-07-14	Endeavor Air Inc.	9E	JFK	PHL	45
246207	4218	2013-08-12	Endeavor Air Inc.	9E	JFK	PHL	45
13516	2889	2013-11-02	Mesa Airlines Inc.	YV	LGA	PHL	46
31496	2889	2013-12-07	Mesa Airlines Inc.	YV	LGA	PHL	46
53621	3793	2013-08-24	Endeavor Air Inc.	9E	JFK	PHL	47

Requerimiento 3

```
def req_3(catalog, carrier, dest, rango_dist):
    """
    Retorna el resultado del requerimiento 3
    """
    # TODO: Modificar el requerimiento 3
    start= get_time()
    filtrado=0
    filtro=lt.new_list()
    if mp.get(catalog["aerolinea"], carrier) is not None:
        mapa= mp.get(catalog["aerolinea"], carrier)
        if mp.get(mapa, dest) is not None:
            lista= mp.get(mapa, dest)
            for i in range (sl.size(lista)):
                viaje= sl.get_element(lista, i)
                if rango_dist[0]<= viaje["distance"] <= rango_dist[1]:
                    lt.add_last(filtro, viaje)
                    filtrado+=1
    filtro= merge_sort(filtro, compare_viajes)
    end= get_time()
    tiempo= delta_time(start,end)
    return tiempo, filtrado, filtro

def merge_sort(my_list, sort_crit):
```

```

def merge_sort(my_list, sort_crit):
    my_list["size"] = len(my_list["elements"])
    if my_list["size"] <= 1:
        return my_list
    mitad = my_list["size"] // 2
    # sub_list(pos, num) → num = cantidad de elementos, no fin
    izq = lt.sub_list(my_list, 0, mitad)
    dere = lt.sub_list(my_list, mitad, my_list["size"] - mitad)
    mitad_izqui = merge_sort(izq, sort_crit)
    mitad_dere = merge_sort(dere, sort_crit)
    return merge(mitad_izqui, mitad_dere, sort_crit)

```

```

def compare_viajes(v1, v2):
    if v1["distance"] < v2["distance"]:
        return True
    elif v1["distance"] > v2["distance"]:
        return False
    else:
        # Combinar fecha y hora en un solo string
        fecha1 = v1["date"] + " " + v1["arr_time"]
        fecha2 = v2["date"] + " " + v2["arr_time"]
        return fecha1 < fecha2

```

Descripción

Entrada

- Código del aeropuerto de destino a analizar (por ejemplo: “JFK”).
- Rango de minutos de antelación en la llegada a filtrar (por ejemplo: [10,30]).

Salidas

- Tiempo de la ejecución del requerimiento en milisegundos.
- Número total de vuelos que cumplen con el filtro del aeropuerto y del rango de antelación.
- Teniendo en cuenta los vuelos que cumplen el filtro, presente los siguientes datos:

- o ID del vuelo.
- o Código del vuelo.
- o Fecha.
- o Nombre de la aerolínea
- o Código de la aerolínea.
- o Aeropuerto de origen. o Aeropuerto de destino.
- o Minutos de antelación en la salida.

Implementado (Sí/No)

Si, por Juan Andrés Lozada Barragán

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
mp.get(catalog["aerolinea"], carrier) y mp.get(mapa, dest)	O(1)
sl.size(lista)	O(1)
for i in range(sl.size(lista))	O(n)

Comparación de rangos y adición a lista	$O(1)$
merge_sort(filtro, compare_viajes)	$O(m \log m)$
delta_time() y retornos	$O(1)$
TOTAL	$O(n+m \log m)$

Análisis

El algoritmo busca los vuelos de una aerolínea hacia un destino específico dentro del catálogo. Luego filtra los vuelos cuya distancia esté dentro de un rango dado y los almacena en una lista temporal. Una vez filtrados, ordena los vuelos por distancia y, en caso de empate, por fecha y hora de llegada usando **merge sort**. Finalmente, devuelve el tiempo de ejecución, la cantidad filtrada y la lista ordenada.

Pruebas (Al menos 5)

Hechos con large:

```
Seleccione una opción para continuar
3
Ingrese el código de la aerolínea (por ejemplo: 'AA', 'UA'): AA
Ingrese el código del aeropuerto de destino (por ejemplo: 'JFK', 'LAX'): LAX
Ingrese la distancia mínima (en millas): 2400
Ingrese la distancia máxima (en millas): 2500
Total de vuelos encontrados: 3350
Tiempo de ejecución [ms]:
466.814
```

ID	Código vuelo	Fecha	Aerolínea	Código	Origen	Destino	Distancia
253087	119	2013-01-02	American Airlines Inc.	AA	EWR	LAX	2454
298753	119	2013-01-03	American Airlines Inc.	AA	EWR	LAX	2454
237279	119	2013-01-04	American Airlines Inc.	AA	EWR	LAX	2454
173443	119	2013-01-05	American Airlines Inc.	AA	EWR	LAX	2454
220456	119	2013-01-06	American Airlines Inc.	AA	EWR	LAX	2454
138058	1	2013-12-31	American Airlines Inc.	AA	JFK	LAX	2475
94190	3	2013-12-31	American Airlines Inc.	AA	JFK	LAX	2475
266157	117	2013-12-31	American Airlines Inc.	AA	JFK	LAX	2475
257482	133	2013-12-31	American Airlines Inc.	AA	JFK	LAX	2475
83350	181	2013-12-31	American Airlines Inc.	AA	JFK	LAX	2475

Ingrese el código de la aerolínea (por ejemplo: 'AA', 'UA'): EV
 Ingrese el código del aeropuerto de destino (por ejemplo: 'JFK', 'LAX'): BWI
 Ingrese la distancia mínima (en millas): 160
 Ingrese la distancia máxima (en millas): 170
 Total de vuelos encontrados: 315
 Tiempo de ejecución [ms]:
 7.55

ID	Código vuelo	Fecha	Aerolínea	Código	Origen	Destino	Distancia
87019	4340	2013-01-01	ExpressJet Airlines Inc.	EV	EWR	BWI	169
235519	4103	2013-01-01	ExpressJet Airlines Inc.	EV	EWR	BWI	169
140111	4485	2013-01-02	ExpressJet Airlines Inc.	EV	EWR	BWI	169
91304	4340	2013-01-02	ExpressJet Airlines Inc.	EV	EWR	BWI	169
36785	4280	2013-01-03	ExpressJet Airlines Inc.	EV	EWR	BWI	169
82883	4625	2013-12-29	ExpressJet Airlines Inc.	EV	EWR	BWI	169
40531	4234	2013-12-30	ExpressJet Airlines Inc.	EV	EWR	BWI	169
286066	4249	2013-12-30	ExpressJet Airlines Inc.	EV	EWR	BWI	169
50636	4625	2013-12-30	ExpressJet Airlines Inc.	EV	EWR	BWI	169
288766	4234	2013-12-31	ExpressJet Airlines Inc.	EV	EWR	BWI	169

Seleccione una opción para continuar

3

Ingrese el código de la aerolínea (por ejemplo: 'AA', 'UA'): UA
 Ingrese el código del aeropuerto de destino (por ejemplo: 'JFK', 'LAX'): ORD
 Ingrese la distancia mínima (en millas): 700
 Ingrese la distancia máxima (en millas): 800
 Total de vuelos encontrados: 6247

Tiempo de ejecución [ms]:

2041.162

ID	Código vuelo	Fecha	Aerolínea	Código	Origen	Destino	Distancia
27068	1676	2013-01-01	United Air Lines Inc.	UA	EWR	ORD	719
262565	544	2013-01-01	United Air Lines Inc.	UA	EWR	ORD	719
81994	580	2013-01-01	United Air Lines Inc.	UA	EWR	ORD	719
291768	985	2013-01-01	United Air Lines Inc.	UA	EWR	ORD	719
3039	32	2013-01-01	United Air Lines Inc.	UA	EWR	ORD	719
120804	689	2013-12-29	United Air Lines Inc.	UA	LGA	ORD	733
274845	331	2013-12-30	United Air Lines Inc.	UA	LGA	ORD	733
183671	687	2013-12-30	United Air Lines Inc.	UA	LGA	ORD	733
284250	691	2013-12-30	United Air Lines Inc.	UA	LGA	ORD	733
91079	331	2013-12-31	United Air Lines Inc.	UA	LGA	ORD	733

Seleccione una opción para continuar

3

Ingrese el código de la aerolínea (por ejemplo: 'AA', 'UA'): 9E

Ingrese el código del aeropuerto de destino (por ejemplo: 'JFK', 'LAX'): PIT

Ingrese la distancia mínima (en millas): 300

Ingrese la distancia máxima (en millas): 400

Total de vuelos encontrados: 718

Tiempo de ejecución [ms]:

41.381

ID	Código vuelo	Fecha	Aerolínea	Código	Origen	Destino	Distancia
272860	4014	2013-05-26	Endeavor Air Inc.	9E	LGA	PIT	335
248592	3580	2013-06-23	Endeavor Air Inc.	9E	LGA	PIT	335
261310	4023	2013-07-05	Endeavor Air Inc.	9E	LGA	PIT	335
94314	3496	2013-09-04	Endeavor Air Inc.	9E	LGA	PIT	335
173679	3311	2013-09-06	Endeavor Air Inc.	9E	LGA	PIT	335
290194	2935	2013-12-28	Endeavor Air Inc.	9E	JFK	PIT	340
55005	2931	2013-12-29	Endeavor Air Inc.	9E	JFK	PIT	340
170845	2935	2013-12-29	Endeavor Air Inc.	9E	JFK	PIT	340
142915	2935	2013-12-30	Endeavor Air Inc.	9E	JFK	PIT	340
244536	2931	2013-12-31	Endeavor Air Inc.	9E	JFK	PIT	340

Seleccione una opción para continuar

3

Ingrese el código de la aerolínea (por ejemplo: 'AA', 'UA'): WN

Ingrese el código del aeropuerto de destino (por ejemplo: 'JFK', 'LAX'): MDW

Ingrese la distancia mínima (en millas): 700

Ingrese la distancia máxima (en millas): 800

Total de vuelos encontrados: 3713

Tiempo de ejecución [ms]:

807.069

ID	Código vuelo	Fecha	Aerolínea	Código	Origen	Destino	Distancia
94844	3848	2013-01-01	Southwest Airlines Co.	WN	EWR	MDW	711
15488	133	2013-01-01	Southwest Airlines Co.	WN	EWR	MDW	711
114172	4105	2013-01-01	Southwest Airlines Co.	WN	EWR	MDW	711
170602	3384	2013-01-01	Southwest Airlines Co.	WN	EWR	MDW	711
161211	195	2013-01-01	Southwest Airlines Co.	WN	EWR	MDW	711
286945	1360	2013-12-31	Southwest Airlines Co.	WN	LGA	MDW	725
206750	824	2013-12-31	Southwest Airlines Co.	WN	LGA	MDW	725
84954	1079	2013-12-31	Southwest Airlines Co.	WN	LGA	MDW	725
28377	3	2013-12-31	Southwest Airlines Co.	WN	LGA	MDW	725
262600	1104	2013-12-31	Southwest Airlines Co.	WN	LGA	MDW	725

Requerimiento 4

```
def req_4(catalog,rango_fecha_ini, rango_fecha_fin, franja_hora_salida_uno, franja_hora_salida_dos, cant_aereolineas_mas_vuelos):
    """
    Retorna el resultado del requerimiento 4
    """

    start= get_time()
    # 1 Filtrar los vuelos por rango de fechas y franja horaria de salida
    rango_fecha_ini = dt.datetime.strptime(rango_fecha_ini, "%Y-%m-%d")
    rango_fecha_fin = dt.datetime.strptime(rango_fecha_fin, "%Y-%m-%d")
    resultado = lt.new_list()

    if rbt.get (catalog["viajes"],rango_fecha_ini) is not None or rbt.get (catalog["viajes"],rango_fecha_fin) is not None:
        list_por_fecha = rbt.values(catalog["viajes"], rango_fecha_ini, rango_fecha_fin)
        list_por_franja = lt.new_list()
        for i in range(sl.size(list_por_fecha)):
            lista_dia = sl.get_element(list_por_fecha, i)
            for j in range(sl.size(lista_dia)):
                vuelo = sl.get_element(lista_dia, j)
                if franja_hora_salida_uno <= vuelo["sched_dep_time"] <= franja_hora_salida_dos:
                    lt.add_last(list_por_franja, vuelo)
        # 2 Agrupar los vuelos por aerolínea
        aereolineas = {}
        for i in range(lt.size(list_por_franja)):
            vuelo = lt.get_element(list_por_franja, i)
            codigo = vuelo["carrier"]
            if codigo in aereolineas:
                aereolineas[codigo]["count"] += 1
                aereolineas[codigo]["suma_duracion"] += int(vuelo["air_time"])
                aereolineas[codigo]["suma_distancia"] += int(vuelo["distance"])
                lt.add_last(aereolineas[codigo]["lista_vuelos"], vuelo)
            else:
                aereolineas[codigo] = {
                    "count": 1,
                    "suma_duracion": int(vuelo["air_time"]),
                    "suma_distancia": int(vuelo["distance"]),
                    "lista_vuelos": lt.new_list()
                }
                lt.add_last(aereolineas[codigo]["lista_vuelos"], vuelo)

        # 3 Insertar las aerolíneas en un heap (máximo por cantidad de vuelos)
        aereo = pq.new_heap(is_min_pq=False)
        for codigo in aereolineas:
            pq.insert(aereo, (aereolineas[codigo]["count"],codigo), codigo)
        # 4 Extraer las top N aerolíneas
        resultado = lt.new_list()
        n = min(cant_aereolineas_mas_vuelos, pq.size(aereo))
        for _ in range(n):
            codigo = pq.remove(aereo)
            info = aereolineas[codigo]

            # 5 Calcular promedios
            prom_duracion = info["suma_duracion"] / info["count"]
            prom_distancia = info["suma_distancia"] / info["count"]

            # 6 Encontrar vuelo con menor duración
            menor = None
            dur_menor = 999999
            fecha_menor = None
```

```

for j in range(lt.size(info["lista_vuelos"])):
    vuelo = lt.get_element(info["lista_vuelos"], j)
    dur = int(vuelo["air_time"])
    fecha_hora = dt.datetime.strptime(
        vuelo["date"] + " " + vuelo["sched_dep_time"], "%Y-%m-%d %H:%M"
    )

    if dur < dur_menor or (dur == dur_menor and fecha_menor < fecha_hora):
        menor = vuelo
        dur_menor = dur
        fecha_menor = fecha_hora

# 7 Agregar al resultado
lt.add_last(resultado, {
    "Aerolínea": codigo,
    "Vuelos totales": info["count"],
    "Duración promedio": round(prom_duracion, 2),
    "Distancia promedio": round(prom_distancia, 2),
    "Vuelo menor duración": {
        "ID": menor["id"],
        "Código": menor["flight"],
        "Fecha salida": menor["date"] + " " + menor["sched_dep_time"],
        "Origen": menor["origin"],
        "Destino": menor["dest"],
        "Duración": menor["air_time"]
    }
})
end= get_time()
tiempo= delta_time(start,end)
return tiempo, pq.size(aereo), resultado

```

Descripción

Entrada	<ul style="list-style-type: none"> • Código de la aerolínea a analizar (por ejemplo: “UA”). • Rango de minutos de retraso en salida a filtrar (por ejemplo: [10,30])
Salidas	<ul style="list-style-type: none"> -Tiempo de la ejecución del requerimiento en milisegundos. -Número total de vuelos que cumplen con el filtro de la aerolínea y rango de retraso. -Teniendo en cuenta los vuelos que cumplen el filtro, presente los siguientes datos: <ul style="list-style-type: none"> • o ID del vuelo. • o Código del vuelo. • o Fecha. • o Nombre de la aerolínea. • o Código de la aerolínea. • o Aeropuerto de origen. • o Aeropuerto de destino. • o Minutos de retraso en la salida.
Implementado (Sí/No)	Si, por Juan Andres Lozada Barragan

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Filtrar los vuelos por rango de fechas y horas	O(n)

Agrupar los vuelos por aerolínea y acumular valores	$O(n)$
Insertar las aerolíneas en un heap limitado por cantidad de valores	$O(k \log k)$
Calcular promedios y hallar la aerolínea con menor duración	$O(k)$
Cálculo final y retorno de resultados	$O(1)$
TOTAL	$O(n + k \log k)$

Análisis

El algoritmo analiza los vuelos en un rango de fechas y horas, agrupándolos por aerolínea para calcular promedios de duración y distancia. Luego usa un heap para obtener las aerolíneas con más valores registrados y determinar la de menor duración promedio. Recorre todos los vuelos una vez ($O(n)$), realiza operaciones logarítmicas en el heap ($O(k \log k)$) y devuelve estadísticas resumidas. Es eficiente, ya que su costo principal crece linealmente con el número de vuelos.

Pruebas (Al menos 5)

Hechos con large:

```
Ingrese la fecha inicial (YYYY-MM-DD): 2013-11-23
Ingrese la fecha final (YYYY-MM-DD): 2013-11-25
Ingrese la hora inicial (HH:MM): 10:00
Ingrese la hora final (HH:MM): 11:00
Ingrese la cantidad de aerolíneas a mostrar (N): 5
```

```
Tiempo de ejecución: ms59.753
Aerolíneas encontradas: 7
```

Aerolínea	Vuelos totales	Duración promedio	Distancia promedio
EV	30	100.4	600.57
DL	30	189.4	1286.63
US	21	136.52	884.14
UA	19	233.95	1615.68
B6	18	145.94	981

== Vuelo con menor duración por aerolínea ==

Aerolínea	ID vuelo	Código	Fecha salida	Origen	Destino	Duración
EV	22761	5346	2013-11-25 10:57	LGA	ROC	43
DL	199479	1869	2013-11-24 11:00	LGA	DTW	83
US	239316	2142	2013-11-23 10:00	LGA	BOS	32
UA	69123	654	2013-11-24 10:59	EWR	ORD	117
B6	172227	518	2013-11-23 10:25	JFK	BOS	34

Ingrese la fecha inicial (YYYY-MM-DD): 2013-05-01

Ingrese la fecha final (YYYY-MM-DD): 2013-05-03

Ingrese la hora inicial (HH:MM): 05:00

Ingrese la hora final (HH:MM): 10:00

Ingrese la cantidad de aerolíneas a mostrar (N): 5

Tiempo de ejecución: ms146.635

Aerolíneas encontradas: 9

== Aerolíneas con mayor número de vuelos ==

Aerolínea	Vuelos totales	Duración promedio	Distancia promedio
UA	147	208.63	1576
EV	134	88.16	597.3
B6	121	146.43	1084.24
DL	118	169.47	1258.88
AA	89	178.42	1329.25

== Vuelo con menor duración por aerolínea ==

Aerolínea	ID vuelo	Código	Fecha salida	Origen	Destino	Duración
UA	118472	428	2013-05-01 08:22	EWR	BOS	37
EV	78140	3822	2013-05-01 07:39	EWR	BDL	23
B6	16776	1010	2013-05-01 06:49	JFK	BOS	35
DL	119170	2179	2013-05-01 08:55	EWR	DTW	73
AA	296968	1838	2013-05-01 08:20	JFK	BOS	34

==== Requerimiento 4 ===

Identificar las N aerolíneas con mayor número de vuelos y su vuelo de menor duración.

Ingrese la fecha inicial (YYYY-MM-DD): 2013-08-15

Ingrese la fecha final (YYYY-MM-DD): 2013-09-15

Ingrese la hora inicial (HH:MM): 08:00

Ingrese la hora final (HH:MM): 10:00

Ingrese la cantidad de aerolíneas a mostrar (N): 4

Tiempo de ejecución: ms697.978

Aerolíneas encontradas: 9

==== Aerolíneas con mayor número de vuelos ===

Aerolínea	Vuelos totales	Duración promedio	Distancia promedio
UA	708	216.64	1667.1
DL	659	161.05	1216.36
EV	655	92.26	621.55
B6	640	150.99	1126.94

==== Vuelo con menor duración por aerolínea ===

Aerolínea	ID vuelo	Código	Fecha salida	Origen	Destino	Duración
UA	135888	1199	2013-08-18 08:07	EWR	BOS	34
DL	200912	867	2013-08-15 09:15	JFK	BOS	31
EV	96018	5811	2013-09-06 08:35	EWR	PVD	29
B6	63197	2280	2013-09-06 08:36	EWR	BOS	36

Ingrese la fecha inicial (YYYY-MM-DD): 2013-10-15
Ingrese la fecha final (YYYY-MM-DD): 2013-11-13
Ingrese la hora inicial (HH:MM): 04:00
Ingrese la hora final (HH:MM): 15:00
Ingrese la cantidad de aerolíneas a mostrar (N): 7

Tiempo de ejecución: ms820.382

Aerolíneas encontradas: 9

== Aerolíneas con mayor número de vuelos ==

Aerolínea	Vuelos totales	Duración promedio	Distancia promedio
UA	2791	220.18	1563.3
EV	2509	91.84	556.19
B6	2206	144.7	1003.99
DL	2050	167.06	1167.33
AA	1427	181.59	1268.17
US	1149	94.44	586.1
MQ	1070	94.78	587.09

== Vuelo con menor duración por aerolínea ==

Aerolínea	ID vuelo	Código	Fecha salida	Origen	Destino	Duración
UA	192941	1686	2013-10-23 06:30	EWR	BOS	32
EV	35264	3847	2013-10-19 07:38	EWR	BDL	24
B6	154075	318	2013-10-23 06:55	JFK	BOS	31
DL	253303	1164	2013-11-02 14:55	JFK	BOS	30
AA	42764	178	2013-10-15 12:50	JFK	BOS	32
US	247521	1909	2013-11-11 06:00	LGA	PHL	27
MQ	256845	3230	2013-10-15 11:05	JFK	DCA	37

== Requerimiento 4 ==

Identificar las N aerolíneas con mayor número de vuelos y su vuelo de menor duración.

Ingrese la fecha inicial (YYYY-MM-DD): 2013-11-16

Ingrese la fecha final (YYYY-MM-DD): 2013-11-20

Ingrese la hora inicial (HH:MM): 04:00

Ingrese la hora final (HH:MM): 08:00

Ingrese la cantidad de aerolíneas a mostrar (N): 5

Tiempo de ejecución: ms199.792

Aerolíneas encontradas: 6

== Aerolíneas con mayor número de vuelos ==

Aerolínea	Vuelos totales	Duración promedio	Distancia promedio
UA	124	214.77	1506.57
B6	100	142.81	1021.4
DL	77	174.62	1213.58
AA	71	190.27	1324.7
EV	68	80.4	468.34

== Vuelo con menor duración por aerolínea ==

Aerolínea	ID vuelo	Código	Fecha salida	Origen	Destino	Duración
UA	217744	215	2013-11-20 06:35	EWR	BOS	37
B6	84754	318	2013-11-17 06:50	JFK	BOS	31
DL	283900	328	2013-11-17 06:00	EWR	DTW	80
AA	1465	307	2013-11-20 07:59	LGA	ORD	116
EV	188764	4114	2013-11-16 07:30	EWR	BOS	35

Requerimiento 5

```
def get_rango(tiempo, rango_aeropuerto, resultado):
    """"
    Retorna el resultado del requerimiento 5
    """
    # TODO: Modificar el requerimiento 5
    start = get_time()

    fecha_ini = dt.datetime.strptime(rango_fecha[0], "%Y-%m-%d")
    fecha_fin = dt.datetime.strptime(rango_fecha[1], "%Y-%m-%d")
    lista_fechas = rbt.values(catalog["vuelos"], fecha_ini, fecha_fin)
    mapa_carrier = np.new_map(2848, 0.5)

    i = 0
    while i < sl.size(lista_fechas):
        dia = sl.get_element(lista_fechas, i)

        j = 0
        while j < sl.size(dia):
            vuelo = dia.get_element(j)
            if vuelo["dest"] == dest:
                dif = diferencia_de_tiempo(vuelo["arr_time"], vuelo["sched_arr_time"])
                if dif < 0:
                    dif = -dif
                air_t = int(vuelo.get("air_time", 0)) if vuelo.get("air_time", "") != "" else 0
                distance = float(vuelo.get("distance", 0)) if vuelo.get("distance", "") != "" else 0.0
                carrier = vuelo["carrier"]
                info = np.get(mapa_carrier, carrier)
                if info is None:
                    info = {
                        "carrier": carrier,
                        "count": 1,
                        "total_delay": dif,
                        "total_air_time": air_t,
                        "total_distance": distance,
                        "max_flight": {
                            "id": vuelo["id"],
                            "flight": vuelo["flight"],
                            "date": vuelo["date"],
                            "dep_time": vuelo["dep_time"],
                            "origin": vuelo["origin"],
                            "dest": vuelo["dest"],
                            "air_time": air_t,
                            "distance": distance
                        }
                    }
                    np.put(mapa_carrier, carrier, info)
                else:
                    info["count"] += 1
                    info["total_delay"] += dif
                    info["total_air_time"] += air_t
                    info["total_distance"] += distance
                    if dif > info["max_flight"]["distance"]:
                        info["max_flight"] = {
                            "id": vuelo["id"],
                            "flight": vuelo["flight"],
                            "date": vuelo["date"],
                            "dep_time": vuelo["dep_time"],
                            "origin": vuelo["origin"],
                            "dest": vuelo["dest"],
                            "air_time": air_t,
                            "distance": distance
                        }
                    np.put(mapa_carrier, carrier, info)
            j += 1
        i += 1

    resultado = rbt.new_map()
    claves = np.key_set(mapa_carrier)
    total_aerolineas = lt.size(claves)

    k = 0
    while k < total_aerolineas:
        carrier = lt.get_element(claves, k)
        info = np.get(mapa_carrier, carrier)
        if info["count"] > 0:
            promedio_delay = info["total_delay"] / info["count"]
            promedio_air_time = info["total_air_time"] / info["count"]
            promedio_distance = info["total_distance"] / info["count"]
            datos = {
                "Aerolinea": carrier,
                "Vuelos totales": info["count"],
                "Retraso promedio llegada (min)": round(promedio_delay, 2),
                "Duración promedio vuelo (min)": round(promedio_air_time, 2),
                "Distancia promedio vuelo (millas)": round(promedio_distance, 2),
                "Vuelo mayor distancia": info["max_flight"]
            }
            clave = int(round(promedio_delay))
            lst = rbt.get(resultado, clave)
            if lst is None:
                lst = lt.new_list()
                lt.add_last(lst, datos)
                rbt.put(resultado, clave, lst)
            else:
                lt.add_last(lst, datos)
                rbt.put(resultado, clave, lst)
        k += 1

    end = get_time()
    tiempo = delta_time(start, end)
    return tiempo, total_aerolineas, resultado
```

Descripción

Este requerimiento toma datos de vuelos dentro de un rango de fechas y destino específico y los agrupa por aerolinea y los ordena por el retraso promedio de llegada.

Entrada

Salidas

- Código del aeropuerto de destino a analizar
- Rango de minutos de anticipio en la llegada a filtrar
- Tiempo de ejecución
- Número total de vuelos que cumplen con el filtro del aeropuerto y del rango de anticipio

- Vuelos que cumplen el filtro:
 - Id
 - Código
 - Fecha
 - Nombre
 - Código aero
 - Aeropuerto origen
 - Aeropuerto destino
 - Minutos antílope en salida

Implementado (Sí/No) Si, por Maria Clara Quijano

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Vuelos dentro de rango del RBT	$O(d+N)$
Inicialización/búsqueda/Inserción hash map	$O(1)$
Bucle sobre días	$O(d)$
<i>Filtro por destino</i>	$O(1)$
<i>Obtener aerolíneas únicas y bucle</i>	$O(A)$
<i>Búsqueda/put rbt</i>	$O(\log R)$
TOTAL	$O(N+A*\log R)$

Análisis

Se usa un hash map para tener mayor eficiencia al agregar datos por su complejidad promedio de $O(1)$.

Se usa un RBT para ordenar automáticamente los datos por el retraso promedio, y tiene una complejidad de $O(\log R)$.

Pruebas (Al menos 5)

Hechos con large:

Rango de fechas (YYYY-MM-DD, YYYY-MM-DD): 2013-01-01, 2013-01-07 Código de aeropuerto destino (ej.: JFK): LAX Cantidad de aerolíneas a mostrar (N): 5 ==== Requerimiento 5: Aerolíneas más puntuales (llegada) ==== Tiempo de ejecución (ms): 1171.28 Aerolíneas consideradas (M): 5 --- Top 5 aerolíneas --																																																																														
<table border="1"> <thead> <tr> <th>Carrier</th><th>Vuelos totales</th><th>Retraso prom. (min)</th><th>Duración prom. (min)</th><th>Dist. prom. (mi)</th><th>Max ID</th><th>Max Vuelo</th><th>Max Fecha</th><th>Max Dep</th><th>Max Origen</th><th>Max Destino</th><th>Max AirTime (min)</th><th>Max Dist (mi)</th></tr> </thead> <tbody> <tr> <td>VX</td><td>41</td><td>36.61</td><td>335.07</td><td>2475</td><td>39354</td><td>407</td><td>2013-01-06</td><td>08:58</td><td>JFK</td><td>LAX</td><td>348</td><td>2475</td></tr> <tr> <td>AA</td><td>62</td><td>59.95</td><td>333.4</td><td>2472,97</td><td>20990</td><td>21</td><td>2013-01-06</td><td>18:56</td><td>JFK</td><td>LAX</td><td>320</td><td>2475</td></tr> <tr> <td>B6</td><td>33</td><td>97.64</td><td>327.15</td><td>2475</td><td>26566</td><td>677</td><td>2013-01-06</td><td>28:36</td><td>JFK</td><td>LAX</td><td>332</td><td>2475</td></tr> <tr> <td>UA</td><td>75</td><td>112.31</td><td>328.57</td><td>2463.52</td><td>31416</td><td>112</td><td>2013-01-06</td><td>08:38</td><td>JFK</td><td>LAX</td><td>332</td><td>2475</td></tr> <tr> <td>DL</td><td>43</td><td>131.37</td><td>334.63</td><td>2475</td><td>24588</td><td>127</td><td>2013-01-06</td><td>16:57</td><td>JFK</td><td>LAX</td><td>349</td><td>2475</td></tr> </tbody> </table>	Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)	VX	41	36.61	335.07	2475	39354	407	2013-01-06	08:58	JFK	LAX	348	2475	AA	62	59.95	333.4	2472,97	20990	21	2013-01-06	18:56	JFK	LAX	320	2475	B6	33	97.64	327.15	2475	26566	677	2013-01-06	28:36	JFK	LAX	332	2475	UA	75	112.31	328.57	2463.52	31416	112	2013-01-06	08:38	JFK	LAX	332	2475	DL	43	131.37	334.63	2475	24588	127	2013-01-06	16:57	JFK	LAX	349	2475
Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)																																																																		
VX	41	36.61	335.07	2475	39354	407	2013-01-06	08:58	JFK	LAX	348	2475																																																																		
AA	62	59.95	333.4	2472,97	20990	21	2013-01-06	18:56	JFK	LAX	320	2475																																																																		
B6	33	97.64	327.15	2475	26566	677	2013-01-06	28:36	JFK	LAX	332	2475																																																																		
UA	75	112.31	328.57	2463.52	31416	112	2013-01-06	08:38	JFK	LAX	332	2475																																																																		
DL	43	131.37	334.63	2475	24588	127	2013-01-06	16:57	JFK	LAX	349	2475																																																																		
<table border="1"> <thead> <tr> <th>Carrier</th><th>Vuelos totales</th><th>Retraso prom. (min)</th><th>Duración prom. (min)</th><th>Dist. prom. (mi)</th><th>Max ID</th><th>Max Vuelo</th><th>Max Fecha</th><th>Max Dep</th><th>Max Origen</th><th>Max Destino</th><th>Max AirTime (min)</th><th>Max Dist (mi)</th></tr> </thead> <tbody> <tr> <td>9E</td><td>2</td><td>23.5</td><td>109</td><td>768</td><td>216913</td><td>3856</td><td>2013-01-04</td><td>06:15</td><td>JFK</td><td>ATL</td><td>118</td><td>768</td></tr> <tr> <td>MQ</td><td>41</td><td>37.66</td><td>122.83</td><td>762</td><td>49583</td><td>4661</td><td>2013-01-06</td><td>17:28</td><td>LGA</td><td>ATL</td><td>138</td><td>762</td></tr> <tr> <td>DL</td><td>170</td><td>41.61</td><td>121.59</td><td>757.67</td><td>19090</td><td>1499</td><td>2013-01-06</td><td>16:54</td><td>LGA</td><td>ATL</td><td>124</td><td>762</td></tr> <tr> <td>FL</td><td>48</td><td>59.23</td><td>125.5</td><td>762</td><td>29294</td><td>349</td><td>2013-01-06</td><td>14:14</td><td>LGA</td><td>ATL</td><td>133</td><td>762</td></tr> <tr> <td>EV</td><td>25</td><td>85.04</td><td>128.44</td><td>746</td><td>14802</td><td>4705</td><td>2013-01-06</td><td>17:22</td><td>EWR</td><td>ATL</td><td>139</td><td>746</td></tr> </tbody> </table>	Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)	9E	2	23.5	109	768	216913	3856	2013-01-04	06:15	JFK	ATL	118	768	MQ	41	37.66	122.83	762	49583	4661	2013-01-06	17:28	LGA	ATL	138	762	DL	170	41.61	121.59	757.67	19090	1499	2013-01-06	16:54	LGA	ATL	124	762	FL	48	59.23	125.5	762	29294	349	2013-01-06	14:14	LGA	ATL	133	762	EV	25	85.04	128.44	746	14802	4705	2013-01-06	17:22	EWR	ATL	139	746
Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)																																																																		
9E	2	23.5	109	768	216913	3856	2013-01-04	06:15	JFK	ATL	118	768																																																																		
MQ	41	37.66	122.83	762	49583	4661	2013-01-06	17:28	LGA	ATL	138	762																																																																		
DL	170	41.61	121.59	757.67	19090	1499	2013-01-06	16:54	LGA	ATL	124	762																																																																		
FL	48	59.23	125.5	762	29294	349	2013-01-06	14:14	LGA	ATL	133	762																																																																		
EV	25	85.04	128.44	746	14802	4705	2013-01-06	17:22	EWR	ATL	139	746																																																																		

Selecione una opción para continuar:

5
Rango de fechas (YYYY-MM-DD, YYYY-MM-DD): 2013-01-01, 2013-01-07
Código de aeropuerto destino (e): JFK: 520

Cantidad de aerolíneas a mostrar (N): 5

*** Requerimiento 5: Aerolíneas más puntuales (llegada) ***

Tiempo de ejecución (ms): 777.09

Aerolíneas consideradas (M): 4

-- Top 4 aerolíneas --

Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)
AA	26	59.27	197.15	1598	236888	413	2013-01-06	06:26	JFK	SJU	197	1598
B6	63	61.25	195.05	1599.11	137927	215	2013-01-06	10:59	EWR	SJU	208	1608
DL	19	113.74	195.53	1598	1627	315	2013-01-06	11:22	JFK	SJU	194	1598
UA	19	182.95	196.53	1688	7723	1283	2013-01-06	11:53	EWR	SJU	201	1608

Rango de fechas (YYYY-MM-DD, YYYY-MM-DD): 2013-01-01, 2013-01-07

Código de aeropuerto destino (e): ORD

Cantidad de aerolíneas a mostrar (N): 5

*** Requerimiento 5: Aerolíneas más puntuales (llegada) ***

Tiempo de ejecución (ms): 546.28

Aerolíneas consideradas (M): 5

-- Top 5 aerolíneas --

Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)
B6	12	16	134.83	740	35121	917	2013-01-06	16:06	JFK	ORD	121	740
9E	18	25.94	128.78	740	77889	3521	2013-01-06	08:12	JFK	ORD	126	740
AA	93	43.89	123.87	733.53	167209	1351	2013-01-06	17:43	JFK	ORD	121	740
HQ	49	50.59	119.73	719	145458	3795	2013-01-06	01:01	EWR	ORD	117	719
UA	96	58.45	123.36	724.69	149943	550	2013-01-06	06:58	LGA	ORD	122	733

Rango de fechas (YYYY-MM-DD, YYYY-MM-DD): 2013-01-01, 2013-01-07

Código de aeropuerto destino (e): SFO

Cantidad de aerolíneas a mostrar (N): 5

*** Requerimiento 5: Aerolíneas más puntuales (llegada) ***

Tiempo de ejecución (ms): 373.58

Aerolíneas consideradas (M): 5

-- Top 5 aerolíneas --

Carrier	Vuelos totales	Retraso prom. (min)	Duración prom. (min)	Dist. prom. (mi)	Max ID	Max Vuelo	Max Fecha	Max Dep	Max Origen	Max Destino	Max AirTime (min)	Max Dist (mi)
DL	32	31.47	347.06	2586	167998	31	2013-01-06	00:17	JFK	SFO	348	2586
AA	25	44.72	350.12	2586	1663	59	2013-01-06	07:43	JFK	SFO	368	2586
B6	28	60.45	339.7	2586	109888	645	2013-01-06	19:42	JFK	SFO	339	2586
VX	28	61.89	346.75	2586	80030	23	2013-01-06	10:35	JFK	SFO	355	2586
UA	98	67.43	341.87	2574.33	96864	272	2013-01-06	18:29	JFK	SFO	336	2586

Requerimiento 6

Descripción

Dado un rango de fechas y distancias, determinar las M aerolíneas más estables en su hora de salida, considerando la puntualidad de sus vuelos.

```

def req_6(catalog, rango_f, rango_d, m):
    """
    Retorna el resultado del requerimiento 6
    rango_f: Rango de fechas a analizar (por ejemplo: ["2013-01-01", "2013-03-31"]).
    rango_d: Rango de distancias (en millas) a analizar (por ejemplo: [500, 1500]).
    m = Cantidad M de aerolíneas a mostrar (por ejemplo: M=5).
    """

    # TODO: Modificar el requerimiento 6
    start = get_time()
    aerolineas = mp.new_map(20, 4)
    fechas = rbt.values(catalog["viajes"], rango_f[0], rango_f[1]) #Me da una lista single list
    for i in range(sl.size(fechas)):
        lista_de_viajes = sl.get_element(fechas, i)
        for j in range(sl.size(lista_de_viajes)):
            viaje = sl.get_element(lista_de_viajes, j)
            if rango_d[0] <= viaje["distance"] and viaje["distance"] <= rango_d[1]:
                v = {"Id": viaje["id"],
                      "Código": viaje["flight"],
                      "F-H Salida": viaje["date"]+" "+viaje["dep_time"],
                      "Origen": viaje["origin"],
                      "Destino": viaje["dest"]}
                diferencia = diferencia_tiempo(viaje["dep_time"], viaje["sched_dep_time"])
                aer = mp.get(aerolineas, viaje["carrier"]) #Obtiene el valor, en este caso None
                if aer is None:
                    datos = mp.new_map(2,1)
                    diferencias_indv = lt.new_list()
                    trayectos = lt.new_list()
                    lt.add_last(trayectos, v)
                    lt.add_last(diferencias_indv, diferencia)
                    mp.put(datos, "dif_ind", diferencias_indv)
                    mp.put(datos, "trayectos", trayectos)

                    mp.put(aerolineas, viaje["carrier"], datos)
                    aer = mp.get(aerolineas, viaje["carrier"])
                    lt.add_last(mp.get(aer,"dif_ind"), diferencia)
                    lt.add_last(mp.get(aer, "trayectos"), v)

```

```

def req_6(catalog, rango_f, rango_d, m):
    """...
    # TODO: Modificar el requerimiento 6
    start = get_time()
    aerolineas = mp.new_map(20, 4)
    fechas = rbt.values(catalog["viajes"], rango_f[0], rango_f[1]) #Me da una lista single l
    for i in range(sl.size(fechas)):
        lista_de_viajes = sl.get_element(fechas, i)
        for j in range(sl.size(lista_de_viajes)):
            viaje = sl.get_element(lista_de_viajes, j)
            if rango_d[0] <= viaje["distance"] and viaje["distance"] <= rango_d[1]:
                v = {"Id": viaje["id"],
                      "Código": viaje["tailnum"], #0 ES EL NÚMERO DEL VUELO "flight"
                      "F-H Salida": viaje["date"]+" "+viaje["dep_time"],
                      "Origen": viaje["origin"],
                      "Destino": viaje["dest"]
                }
                diferencia = diferencia_tiempo(viaje["dep_time"], viaje["sched_dep_time"])
                aer = mp.get(aerolineas, viaje["carrier"]) #Obtiene el valor, en este caso
                if aer is None:
                    datos = mp.new_map(2,1)
                    diferencias_indv = lt.new_list()
                    trayectos = lt.new_list()
                    lt.add_last(trayectos, v)
                    lt.add_last(diferencias_indv, diferencia)
                    mp.put(datos, "dif_ind", diferencias_indv)
                    mp.put(datos, "trayectos", trayectos)

                    mp.put(aerolineas, viaje["carrier"], datos)
                else:
                    lt.add_last(mp.get(aer,"dif_ind"), diferencia)
                    lt.add_last(mp.get(aer, "trayectos"), v)
            promedio = 0
            desviacion = 0
            aero = pq.new_heap(True)
            clave=mp.key_set(aerolineas)
            for i in range(mp.size(aerolineas)):
                mapa = mp.get(aerolineas, lt.get_element(clave, i))
                lista = mp.get(mapa, "dif_ind")
                trayectos = mp.get(mapa, "trayectos")
                t = 0
                suma = 0
                for j in range(lt.size(lista)):
                    t += 1
                    suma += lt.get_element(lista, j)
                promedio = suma/t
                t = 0
                suma = 0
                menor = 999999999
                vuelo = None
                for j in range(lt.size(lista)):
                    d=abs(promedio-lt.get_element(lista, j))
                    if d < menor:
                        menor = d
                        vuelo = lt.get_element(trayectos, j)
                    t += 1
                    suma += (lt.get_element(lista, j) - promedio)**2
                desviacion = math.sqrt(suma/t)

                info = {"Aerolínea": lt.get_element(mp.key_set(aerolineas),i), #Código aerolínea,
                        "# vuelos": lt.size(trayectos),
                        "Promedio (min)": round(promedio, 2),
                        "Estabilidad": round(desviacion, 2),
                        "Vuelo (Id, Código de Vuelo, F/H salida, Aerop Origen y Dest)": f'{vuelo["Id"]} | {vuelo["Código"]} | {vuelo["F-H Salida"]} | {vuelo["Origen"]} -> {vuelo["Destino"]}'}
            pq.insert(aero, (desviacion, promedio), info)

```

Entrada

- Rango de fechas a analizar.
- Rango de distancias (en millas) a analizar.
- Cantidad M de aerolíneas a mostrar.

Salidas

Tiempo de la ejecución del requerimiento en milisegundos.
Número total de aerolíneas analizadas (M).

Para cada una de las aerolíneas:

- Código de la aerolínea.
- Número total de vuelos analizados en el rango especificado.
- Promedio del retraso/anticipo en la salida de sus vuelos, en minutos.
- Estabilidad de la salida (desviación estándar del retraso/anticipo en la salida) en minutos.
- Vuelo con el retraso más cercano al promedio:
 - ID del vuelo
 - Código del vuelo
 - Fecha-Hora de salida del vuelo
 - Aeropuerto de origen
 - Aeropuerto de destino

Implementado (Sí/No)

Sí, por Juliana Rodríguez

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Lista de valores según un rango (values)	$O(1)$
Primer recorrido for (lista de listas)	$O(n)$
Obtener elemento de una lista single	$O(n)$
Segundo recorrido for (viajes)	$O(m)$
Obtener elemento dentro de lista single	$O(m)$
Get de un mapa x3	$O(r/R)$
Recorrido del nuevo mapa con los vuelos filtrados	$O(r)$
Recorrido for para hallar promedio	$O(l)$
Recorrido for para hallar desviación	$O(l)$
Insert de cola de prioridad	$O(1+\log r)$
TOTAL	$O(n*m*r/R + r/R*l*(1+\log r))$

Análisis

La función se divide en dos partes, la primera analiza los vuelos que cumplen los criterios y guarda la información de estos y en la segunda parte se realiza el cálculo del promedio y desviación estándar ya que con esto se presentarán los datos (además de ser el propósito de la función).

En la primera parte del árbol con las fechas primero se realiza una extracción de las fechas dentro del rango dado por parámetro, lo que genera una `single_linked`, es decir una lista de listas. Para poder recorrer todas las listas (que son los valores de las fechas) y recorrer todos los trayectos dentro de estas se hizo dos recorridos for (por eso $n \times m$); de allí se analizaron todos los vuelos filtrando por el rango de distancias (pero como este se hizo con un if y llamado de llaves no aporta a mayor complejidad), acá se guardó la información necesaria según lo que pedía el req. Se creó un mapa para poder guardar dos listas, una con la lista de las diferencias entre el tiempo de salida real y programado (para usarlo más adelante en el cálculo del promedio y desviación) y otra con la información del viaje.

En la segunda parte se recorre el mapa antes creado para poder recorrer las listas de las diferencias, así, en el primero for se calculaba el promedio por aerolínea, después se hizo otro for para poder calcular la desviación estándar –La razón de no haber hecho el cálculo en un mismo for para ahorrar complejidad es porque la desviación estándar se calcula por cada viaje y se necesita del promedio. En el mismo se calculó el viaje que tenía menos diferencia con el promedio, pues era parte del req. Finalmente se ajustaron los datos para su presentación y se añadieron en una cola de prioridad para poder organizarlo más fácil.

Pruebas

Hechas con large:

```
Indique el rango de fechas:
Fechas mínima: 2013-11-23
Fecha máxima: 2013-11-25
Indique el rango de distancias:
Distancia mínima: 2400
Distancia máxima: 2500
Diga el número de aerolíneas a mostrar: 5
Tiempo de ejecución: 169.506
Aerolíneas analizadas: 6
```

Aerolínea	# vuelos	Promedio (min)	Estabilidad	Vuelo (Id, Código de Vuelo, F/H salida, Aerop Origen y Dest)
AS	6	-0.33	19.65	11111 N435AS 2013-11-24 18:21 EWR -> SEA
UA	63	5.13	64.49	133993 N36469 2013-11-23 19:44 EWR -> LAX
DL	30	22	99.44	254174 N199DN 2013-11-24 20:37 JFK -> LAX
B6	30	-20.2	114.57	168441 N834JB 2013-11-24 06:54 JFK -> LAX
VX	22	-42	138.22	38335 N641VA 2013-11-23 19:55 JFK -> LAX

```
Indique el rango de fechas:
Fechas mínima: 2013-01-01
Fecha máxima: 2013-03-31
Indique el rango de distancias:
Distancia mínima: 500
Distancia máxima: 1500
Diga el número de aerolíneas a mostrar: 5
Tiempo de ejecución: 8546.112
Aerolíneas analizadas: 11
```

Aerolínea	# vuelos	Promedio (min)	Estabilidad	Vuelo (Id, Código de Vuelo, F/H salida, Aerop Origen y Dest)
00	1	67	0	84090 N978SW 2013-01-30 12:22 LGA -> ORD
9E	1592	27.42	116.03	148212 N933XJ 2013-01-09 19:42 JFK -> ORD
US	1861	-4.85	126.86	69787 N191UW 2013-03-19 06:25 LGA -> CLT
B6	6454	16.08	129.26	163143 N568JB 2013-02-19 08:51 EWR -> MCO
AA	4958	0.91	139.67	121516 N364AA 2013-03-19 17:31 JFK -> MIA

Indique el rango de fechas:

Fechas mínima: 2013-11-20

Fecha máxima: 2013-11-25

Indique el rango de distancias:

Distancia mínima: 10

Distancia máxima: 40

Diga el número de aerolíneas a mostrar: 6

Tiempo de ejecución: 193.946

Aerolíneas analizadas: 0

No hay 6 viajes.

No hay ningún vuelo.

Indique el rango de fechas:

Fechas mínima: 2013-05-15

Fecha máxima: 2013-08-11

Indique el rango de distancias:

Distancia mínima: 19

Distancia máxima: 58

Diga el número de aerolíneas a mostrar: 3

Tiempo de ejecución: 2551.936

Aerolíneas analizadas: 0

No hay 3 viajes.

No hay ningún vuelo.

Indique el rango de fechas:

Fechas mínima: 2013-04-21

Fecha máxima: 2013-11-30

Indique el rango de distancias:

Distancia mínima: 100

Distancia máxima: 500

Diga el número de aerolíneas a mostrar: 2

Tiempo de ejecución: 6188.784

Aerolineas analizadas: 11

Aerolínea	# vuelos	Promedio (min)	Estabilidad	Vuelo (Id, Código de Vuelo, F/H salida, Aerop Origen y Dest)
OO	22	9.95	43.83	61760 N746SK 2013-11-30 16:48 LGA -> IAD
YV	131	47.13	107.45	213555 N522LR 2013-08-21 16:52 LGA -> IAD

Promedio de tiempo: 3530.057 ms