

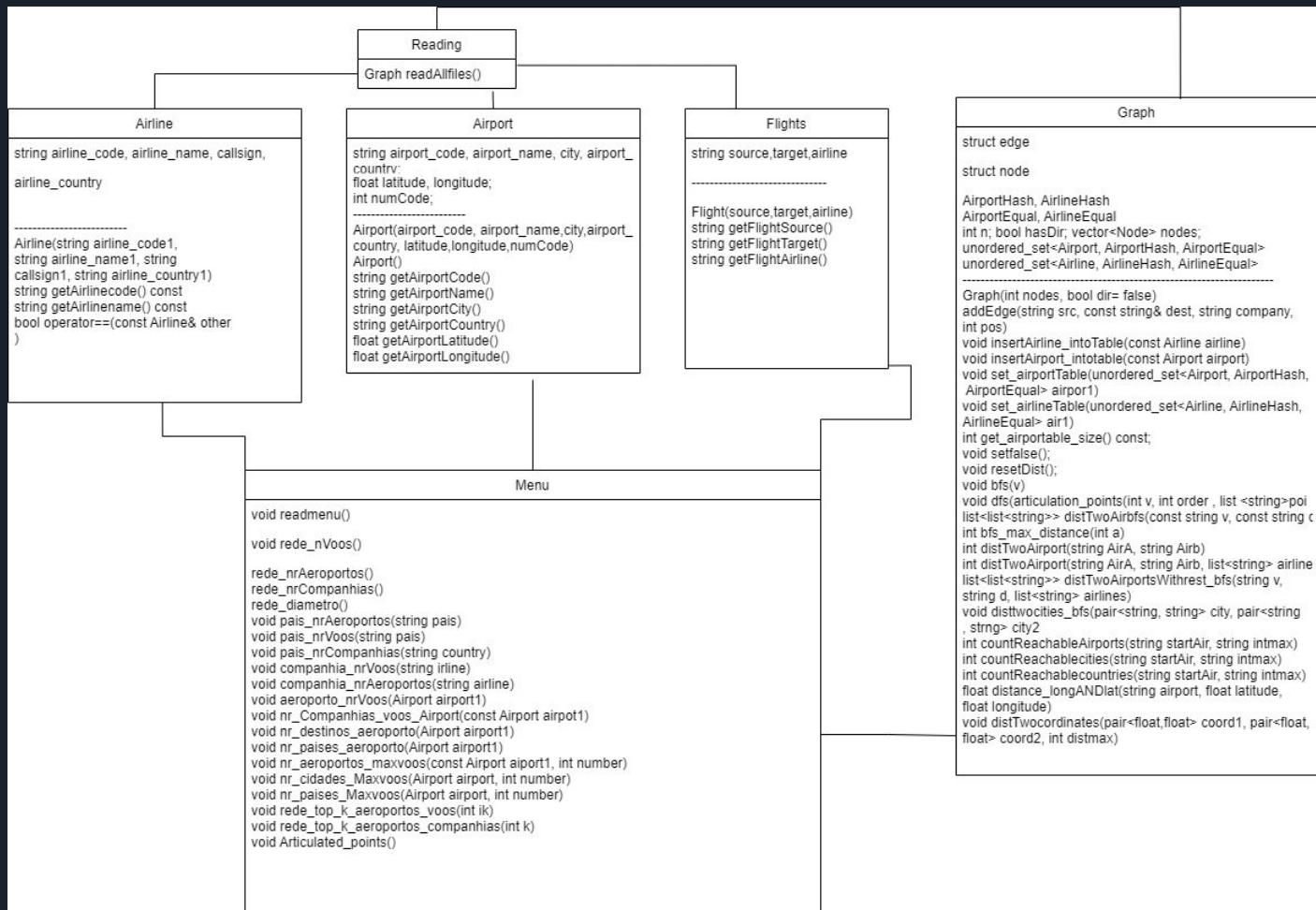


Transportes Aéreos

Trabalho prático 2 de AED 2022/23

Projeto realizado pelo grupo 66:
André Pinto, 202108856
Diana Martins, 202108815
João Oliveira, 202108737

Classes





Leitura do Dataset

A leitura foi feita recorrendo a *getlines* com ‘,’ como delimitador de caracter, na função *readAllFiles()* da classe *Reading*.

airlines.csv

- Criação de um *unordered_set<Airline, AirlineHash, AirlineEqual> airlineTable* que, da leitura do ficheiro, guarda cada companhia da forma: *Airline(AIRLINE_CODE, AIRLINE_NAME, CALLSIGN, AIRLINE_COUNTRY)*, recebendo os *AIRLINE_CODE*, *AIRLINE_NAME*, *CALLSIGN* e *AIRLINE_COUNTRY* do *airlines.csv*.



Leitura do Dataset

airports.csv

- Criação de um *unordered_set*<Airport, AirportHash, AirportEqual> *airportTable* que, da leitura do ficheiro, guarda cada Aeroporto da forma: *Airport(AIRPORT_CODE, AIRPORT_NAME, CITY, AIRPORT_COUNTRY, LATITUDE, LONGITUDE, count)*, recebendo os *AIRPORT_CODE*, *AIRPORT_NAME*, *CITY*, *AIRPORT_COUNTRY*, *LATITUDE*, *LONGITUDE* e *count* do *airlines.csv*. Os códigos dos aeroportos são adicionados ao grafo em que o *count -1* representa a posição do Aeroporto no vetor de nodes.

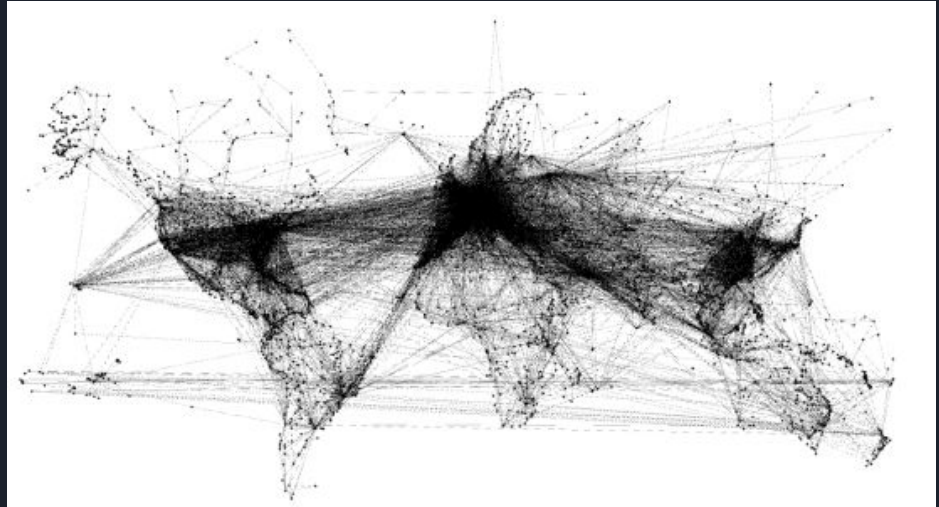
flights.csv

- Criação de um *vector*<Flight> *flights* que, da leitura do ficheiro, guarda cada voo da forma: *Flight(SOURCE, TARGET, AIRLINE)*, recebendo os *SOURCE*, *TARGET* e *AIRLINE* do *flights.csv*. Os voos são depois adicionados a um grafo.

Grafo - Voos

```
struct Edge {  
    string dest;  
    list<string> company;  
};
```

```
struct Node {  
    string src;  
    list<Edge> adj;  
    bool visited;  
    int dist;  
    int low;  
    int num;  
};
```





Tabelas de Hash

- `unordered_set<Airport,AirportHash,AirportEqual>`
- `unordered_set<Airline,AirlineHash,AirlineEqual>`

```
struct AirportHash {
    std::size_t operator()(const Airport& airport) const {
        size_t h = 0;
        h = 37 * hash<string>()(airport.getAirportcode());
        return h;
    }
};

struct AirlineHash{
    std::size_t operator()(const Airline& airline) const {
        std::size_t h = 0;
        h = 37 * h + std::hash<std::string>()(airline.getAirlinename());
        h = 37 * h + std::hash<std::string>()(airline.getAirlinecode());
        h = 37 * h + std::hash<std::string>()(airline.getAirlinecountry());
        h = 37 * h + std::hash<std::string>()(airline.getAirlinecallsign());
        return h;
    }
};

struct AirportEqual {
    bool operator()(const Airport& airport1, const Airport& airport2) const {
        return airport1 == airport2;
    }
};

struct AirlineEqual{
    bool operator()(const Airline& airline1, const Airline& airline2) const {
        return airline1 == airline2;
    }
};
```



Funcionalidades

Origem/Destino - Local

- Saber qual a melhor maneira de voar de um local para o outro.
- Dados os códigos de dois aeroportos;

Calcular as rotas entre dois aeroportos usando o menor número de voos possível.

$(O(|V| + |E|))$

- Dados os nomes de duas cidades;

Calcular as rotas entre duas cidades usando o menor número de voos possível.

$(O(|V| + |E|))$

- Dadas duas coordenadas.

Calcular a menor rota entre aeroportos com uma distância menor que x km das coordenadas dadas. Para saber a distância entre coordenadas foi utilizada a fórmula de Haversine. $(O(|V| + |E|))$



Funcionalidades

Rede de Voos

- Um utilizador pode querer usar qualquer companhia aérea, apenas uma companhia aérea ou uma lista de companhias aéreas para fazer uma viagem entre dois aeroportos.

Calcular as rotas entre dois aeroportos usando o menor número de voos possível se o utilizador não pretender ter nenhum tipo de restrições a nível de companhias. ($O(|V| + |E|)$)

Caso o utilizador pretenda restringir as companhias aéreas a serem utilizadas na sua viagem, os códigos destas são guardadas numa lista. Aquando da escolha das melhores rotas, apenas serão escolhidas viagens realizadas por essas mesmas companhias. ($O(|V| + |E|)$)



Funcionalidades

Informações sobre um Aeroporto

Saber:

- quantos voos existem a partir de um aeroporto

Somar todos os voos (guardados no grafo) associados a um aeroporto ($O(n)$);

- quantas companhias aéreas diferentes estão envolvidas nos voos que partem de um aeroporto;

Guardar os códigos das companhias aéreas associadas a um aeroporto num *set*, imprimindo depois os elementos desse mesmo *set* e o seu tamanho ($O(n)$);



Funcionalidades

Informações sobre um Aeroporto

Saber:

- para quantos aeroportos diferentes se pode ir partindo de um aeroporto;

Somar os destinos dos voos (guardados no grafo) associados a um aeroporto ($O(n)$).

- para quantos países diferentes se pode ir partindo de um aeroporto;

Guardar os nomes dos países de destino (exceto o próprio país) dos voos (guardados no grafo) associadas a um aeroporto num set, imprimindo depois o tamanho desse set ($O(n)$);



Funcionalidades

Informações sobre um Aeroporto

Saber:

- o maior número de aeroportos atingíveis usando um máximo de Y voos;

Usando uma bfs, contar o número de aeroportos que estejam a uma “distância” entre 1 e Y voos do aeroporto em questão ($O(n)$).

- o maior número de cidades atingíveis usando um máximo de Y voos;

Usando uma bfs, adicionar a um set o nome da cidade (guardada na *airportTable*) dos aeroportos que estejam a uma “distância” entre 1 e Y voos do aeroporto em questão, imprimindo o tamanho desse set ($O(n)$).

- o maior número de países atingíveis usando um máximo de Y voos.

Usando uma bfs, adicionar a um set o nome do país (guardado na *airportTable*) dos aeroportos que estejam a uma “distância” entre 1 e Y voos do aeroporto em questão, imprimindo o tamanho desse set ($O(n)$);



Funcionalidades

Tarefas de Valorização

- Quando damos a melhor possibilidade de voo, indicamos todas as possibilidades com o menor número de voos e qual é esse número;
- Possibilitamos a pesquisa de percursos que usem só uma companhia aérea;
- Calcular os pontos de articulação existentes na rede (incluídos nas estatísticas globais da rede, no próximo slide).



Funcionalidades

Tarefas de Valorização

Saber estatísticas globais:

- da rede;

É possível ver: o número de aeroportos (tamanho da airportTable, $O(1)$), de voos (somar os voos de todos os aeroportos, guardados no grafo, $O(n)$) e de companhias (tamanho da airlineTable, $O(1)$), o diâmetro (usando uma bfs, ver qual é a maior distância entre dois *nodes* do grafo, $O(|V|(|V| + |E|))$), os pontos de articulação (usar uma dfs para encontrar os pontos de articulação do grafo, $O(|V|(|V| + |E|))$) e o top-k de aeroportos com mais voos (guardar num set os aeroportos ordenados por número de voos, imprimindo o nome e número de voos apenas dos k primeiros elementos do set, $O(n)$) e mais companhias (guardar num set os aeroportos ordenados por número de companhias, imprimindo o nome e número de voos apenas dos k primeiros elementos do set, $O(n)$), sendo k um valor inserido pelo utilizador.



Funcionalidades

Tarefas de Valorização

Saber estatísticas globais:

- de um país;

É possível ver: o número de aeroportos (somar o número de aeroportos de um país, guardados na airportTable, $O(n)$) de voos (somar o número de voos dos aeroportos de um país, guardados no grafo e na airportTable, $O(n)$) e de companhias (somar o número de companhias de um país, guardadas na airlineTable, $O(n)$).

- de uma companhia.

É possível ver: o número de aeroportos (contar os aeroportos que pertencem a determinada companhia, guardados no grafo, $O(n)$) e de voos (contar os voos associados a uma companhia, $O(n)$).



Interface com o Utilizador

- O menu permite aceder a todas as funcionalidades. Está dividido entre saber qual a melhor forma de chegar de um local ao outro (entre aeroportos, cidades, ou dadas duas coordenadas), obter informações gerais acerca de um aeroporto (desde o número de voos que dele partem até à quantidade de países atingíveis partindo deste aeroporto e usando apenas Y voos) e saber quais as estatísticas globais da rede ou de um país ou companhia.

Interface com o Utilizador

Exemplo Ilustrativo

Insira uma tecla de acordo com o que pretende fazer:

```
1 : Saber qual a melhor maneira de voar de um local para o outro. (Local = aeroporto)
2 : Saber qual a melhor maneira de voar de um local para o outro com restricao de companhia. (Local = aeroporto).
3 : Saber qual a melhor maneira de voar de um local para o outro. (Local = cidade)
4 : Saber qual a melhor maneira de voar de um local para o outro. (Local = conjunto de coordenadas)
5 : Obter informacoes acerca de um aeroporto.
6 : Ver as estatisticas globais.
q : Quit.
```

6

Insira uma tecla de acordo com o que pretende fazer:

```
1 : Saber estatisticas globais da rede.
2 : Saber estatisticas globais de um pais.
3 : Saber estatisticas globais de uma companhia.
q : Quit
```

1

Insira o valor de k para o top-k de aeroportos com mais voos e/ou companhias.

3

```
Esta rede tem um total de 3019 aeroportos.
Esta rede tem um total de 63832 voos.
Esta rede tem um total de 444 companhias.
Esta rede tem um diametro igual a 12.
Existe um total de 309 pontos de articulacao.
Os 3 aeroportos com mais voos sao os seguintes:
ATL com 909 voos.
ORD com 556 voos.
PEK com 526 voos.
Os 3 aeroportos com mais companhias sao os seguintes:
CDG com 102 companhias.
FRA com 97 companhias.
BKK com 96 companhias.
```




Destaque

- Dados dois locais, indicar a melhor maneira de voar de um para o outro (utilizando o menor número de voos possível).



Dificuldades

- Encontrar as melhores rotas;
- Permitir a escolha das companhias aéreas a utilizar.

Esforço de cada elemento:

- André Pinto: 33.3%
- Diana Martins: 33.3%
- João Oliveira: 33.3%