**Corso di Laboratorio di Algoritmi e Strutture Dati**

**REPORT PROGETTO II: COMPAGNIA AEREA**

# Anselmi Gianmarco – N86001394

# Nikolov Aleks Nikolaev – N86003002

**Lama Lorenzo – N86002845**

**Mappa grafo**

Immagine che contiene testo, mappa

Descrizione generata automaticamente

# Implementazione

# La traccia prevede la realizzazione di un programma per la gestione delle prenotazioni di voli.

Prima di essere eseguito, il programma popola l’aeroporto con 20 destinazioni e tratte arbitrarie, riempendo un grafo chiamato *t\_grf* in concomitanza con una struct *t\_arc* per gli adiacenti. Un vertice di tale grafo è dato da una struct contenente il *nome* dell’aeroporto, una variabile *popolarità* usata nella ricerca della destinazione più gettonata, una variabile intera *camminoMinimo* usata nell’algoritmo di Dijkstra per la memorizzazione del peso totale del cammino fino a un dato vertice; un puntatore a *t\_grf* per raggiungere gli altri vertici e un puntatore a *t\_arc* per raggiungere gli adiacenti di un vertice; infine un puntatore a t\_grf chiamato *prev* usato nell’algoritmo di Dijkstra per puntare al vertice che precede nel cammino minimo. Inoltre nella struct *t\_arc* sono presenti i pesi *prezzo* e *durata* utili per la memorizzazione di informazione per i voli.

**Per motivi di semplicità, si assume che la disponibilità dei voli non sia vincolata a date ne al numero dei posti disponibili sul volo stesso.**

Successivamente, un ciclo do-while stampa un menu finchè non si inserisce un input d’uscita (4) con successiva richiesta di conferma di chiusura del programma. Il menu è gestito da uno switch case.

Il programma offre le seguenti funzionalità:

* **Effettua nuova registrazione**  effettuata da un nuovo cliente, implementato dalla funzione ***effettuaRegistrazione***; questa permette la registrazione di un nuovo cliente nel sistema. Si inserisce un nome utente e una password rispettando il fatto che queste ultime non possono contenere spazi e la loro lunghezza non deve superare i 20 caratteri. **Gli input sono case-sensitive**. E’ possibile registrarsi come clienti ma non come admin (si veda scelta 3 per la visualizzazione delle credenziali per un admin)
* **Esegui accesso**, implementata dalla funzione ***eseguiAccesso*** e ***gestisciUtente***.

***eseguiAccesso*** restituisce 0 se il login fallisce, 1 se le credenziali appartengono a un semplice cliente, 2 se invece appartengono a un admin

***gestisciUtente*** mostrerà la schermata menù del cliente o dell’admin in base all’esisto della funzione eseguiAccesso

Da qui distingueremo i 2 casi differentemente:

1. ***Menù cliente:*** da qui un ciclo do while stampa un menu finchè non si inserisce un input di uscita (4). Il menù è gestito da uno switch case:

* **Effettua prenotazione** dalla quale è possibile

**Effettuare una prenotazione inserendo partenza e destinazione:** si inserisce un aeroporto di partenza e uno di destinazione, dopo di che si effettua una breadth first search per controllare che l'aeroporto di destinazione sia raggiungibile da quello di partenza. Se la tratta esiste allora si chiederà di scegliere tra una tratta più economica e una più breve, che verrà trovata con l’applicazione dell’algoritmo di Dijkstra; verrà stampata la tratta desiderata e si chiederà se si vuole applicare uno sconto per l’acquisto. Dopo avere o non avere applicato uno sconto, si richiederà la conferma di prenotazione al cliente; la prenotazione verrà memorizzata in una struttura *t\_prenotazione* all’interno del nodo *t\_utente* dell’utente

**Effettuare una prenotazione inserendo solo partenza**

Si inserisce un aeroporto di partenza, dopo di che si ha la possibilità di scegliere la destinazione più economica e destinazione più gettonata (tramite la visita nell’albero delle precedenti prenotazioni degli altri clienti). Per entrambe le opzioni si ha la possibilità di applicare uno sconto al prezzo del biglietto e successivamente di confermare l’acquisto

**Mostrare i voli**

Mostra tutti i voli, stampati come liste di adiacenza

**Tornare indietro al menù utente**

Torna indietro alla schermata del menù utente

* **Stampa prenotazioni attive**

Stampa tutte le prenotazioni effettuate dall’utente, contenute all’interno delle strutture *t\_utente*

* **Gestisci punti e tickets** dalla quale è possibile

**Vedere i punti disponibili**

Viene stampato il numero dei punti disponibili per l’utente

**Vedere tickets disponibili**

Da qui è possibile visionare i tickets disponibili (punti convertiti in buoni sconto)

**Convertire punti in tickets**

Da qui è possibile convertire i punti posseduti dal cliente in tickets sconto per i voli. Le regole di conversione sono le seguenti:

1 – E’ possibile convertire punti in tickets a patto che il numero di punti sia compreso tra 50 e 250

2 – I tickets di sconto ottenibili vanno dal 5%, 10% fino al 25%

3 – 50 punti corrispondono a uno sconto del 5%, 100 punti del 10%.. fino a 250 che corrispondo a un ticket del 25%

4 – Inserire numeri diversi da 50, 100, 150, 200, 250 è inutile (con 80 punti non otterrete uno sconto dell’8% ma del 5% e vi saranno sottratti solo 50 punti) ma non errato

**Torna indietro al menù utente**

Torna indietro alla schermata del menù utente

1. ***Menù admin:*** da qui un ciclo do while stampa un menu finchè non si inserisce un input di uscita (6). Il menù è gestito da uno switch case.

* ***Aggiungi una nuova destinazione:***

Si chiede di inserire il nome del nuovo aeroporto, che viene inserito nel grafo dei voli

* ***Aggiungi un nuovo volo:***

Permette di aggiungere una nuova tratta. Le richieste da soddisfare sono: partenza, destinazione, prezzo e durata

* ***Rimuovi una destinazione:***

Rimuove una destinazione da un aeroporto

* ***Rimuovi un volo:***

Rimuove un volo dal grafo dei voli

* ***Stampa tutti i voli:***

Stampa il grafo dei voli come una lista di adiacenze

* ***Logout***

Torna indietro al menù utente

**Scelta delle strutture dati**

La struttura dati utilizzata per la registrazione dei voli è un grafo orientato, composto da una struct *t\_grf* per i vertici e da una *t\_arc* per le tratte e la memorizzazione del prezzo e della durata di un volo. Il grafo è stato implementato come una lista di liste, per permettere la facile e dinamica aggiunta/rimozione di aeroporti, e la comoda applicazione di funzioni ricorsive.

La struttura dati utilizzata per la registrazione degli utenti è un albero di ricerca binario, dove ogni nodo presenta una struct *t\_utente*, contenente tutte le informazioni sull’utente, così come una *t\_lista\_P*, contentente le prenotazioni effettuate dal singolo cliente.

**Suddivisione del lavoro tra i componenti del gruppo**

Lama Lorenzo:

Nikolov Aleks Nikolaev: strutture e funzioni per la gestione del grafo dei voli, funzioni per le ricerche sui voli, gestione della partenza da sola destinazione

Anselmi Gianmarco: abbellimento interfaccia, testing, report, mappa grafo e menù admin

**Esempio di esecuzione**