

Corso di Laboratorio di Programmazione

Sistema di segnalazione dei treni

2° assegnamento

Introduzione

Vi è stato commissionato un sistema di segnalazione ferroviaria. Questo sistema serve per descrivere lo stato dei treni e delle stazioni su una linea ferroviaria, e per gestire la circolazione dei treni in arrivo e in partenza dalle varie stazioni.

Linea ferroviaria

La linea ferroviaria è composta da un'unica tratta interrotta da varie stazioni lungo il percorso. Tutta la linea ha due binari per la circolazione di treni in contemporanea lungo i due versi. La posizione di ciascuna stazione è identificata dalla sua distanza dal primo estremo.

La tratta è descritta dall'elenco delle sue stazioni: è presente una stazione all'inizio della linea e la stazione più distante rappresenta la stazione capolinea. La descrizione è presente nel file **line_description.txt**, che deve essere posto nella directory root del progetto. Tale file ha il seguente formato:

```
[nome stazione origine]
[nome stazione] [campo == 0 se stazione principale, == 1 se stazione locale] [distanza dall'origine]
...
[nome stazione] [campo == 0 se stazione principale, == 1 se stazione locale] [distanza dall'origine]
```

(Nota: le parentesi quadrate non devono essere inserite nel file, servono solo per indicare i campi in questo documento).

I nomi delle stazioni sono stringhe che possono contenere più parole; le distanze sono interi che esprimono i km dalla stazione origine. La stazione origine (sempre la prima nel file) si considera fissata al km 0, e non ha l'indicazione della distanza. **La distanza tra una stazione e la successiva è di minimo 20 km. Se il file contiene stazioni a distanza minore, la più lontana dall'origine deve essere ignorata. Dovranno essere ignorati anche i dati relativi a tale stazione nella timetable.**

Stazioni

Esistono due tipologie di stazioni: principali e locali, che devono essere organizzate sfruttando un'interfaccia comune e classi derivate. Ciascuna stazione è in contatto con i treni, che fanno delle segnalazioni quando sono in arrivo, chiedendo su che binario devono transitare ed eventualmente fermarsi. Ciascuna stazione deve rispondere alle segnalazioni dei treni in arrivo fornendo il binario dove il treno deve transitare (ed eventualmente fermarsi). Esistono due tipi di binari:

- Binari standard: permettono sia il solo transito che la salita/discesa dei passeggeri. La velocità massima deve essere limitata a 80 km/h per una lunghezza complessiva di 10 km attorno alla stazione (5 km dalla stazione verso entrambi i sensi di marcia);
- Binari di transito: permettono il solo transito del treno, ma non impongono limiti di velocità.

Le stazioni **secondarie** sono dotate di quattro binari standard (due per ciascun senso di marcia) e due di transito (uno per ciascun senso di marcia), mentre quelle **principali** hanno solamente quattro binari standard (due per senso di marcia).

I treni che si fermano in una stazione stanno fermi 5 minuti per la salita/discesa dei passeggeri; possono rimanere fermi più a lungo se necessario (per esempio, in caso di traffico), ma non possono fermarsi per un tempo inferiore.

Se una stazione riceve una richiesta di transito che non può gestire (perché è piena) può richiedere al treno di arrestarsi in una apposita zona di parcheggio distante 5 km dalla stazione. Questa zona di parcheggio può ospitare infiniti treni. **Il binario si considera unico (per ciascun senso di marcia) fino a questa zona di parcheggio. Da questa zona di parcheggio alla stazione sono presenti un numero di binari pari a quelli standard della stazione.**

Treni

Esistono tre tipologie di treni: regionali, alta velocità, alta velocità super, che devono essere organizzati sfruttando un'interfaccia comune e classi derivate. I treni regionali fermano in tutte le stazioni, quelli alta velocità e alta velocità super fermano solo nelle stazioni principali. Tutti i treni percorrono l'intera tratta, **cioè tutta la linea in un solo verso** (nessuno termina in una stazione intermedia). I treni hanno una velocità massima che dipende dalla tipologia:

- 160 km/h per i treni regionali
- 240 km/h per i treni alta velocità
- 300 km/h per i treni alta velocità super

Ogni treno può decidere la velocità di crociera, rimanendo nei limiti elencati sopra e con un ulteriore limite fissato dalle stazioni. Quando un treno si trova alla distanza di 20 km invia una segnalazione alla prossima stazione, comunicando se deve fermarsi o se deve solo transitare, e richiedendo il binario su cui deve viaggiare.

I treni variano la loro velocità in modo istantaneo. Per esempio, un treno fermo in partenza da una stazione passa istantaneamente da 0 km/h a 80 km/h (valore limite della stazione su binario standard) e, dopo 5 km, passa istantaneamente alla velocità di crociera stabilita. La lunghezza del treno è trascurabile rispetto alle tratte, perciò i limiti di velocità si applicano ai treni modellandoli come punti lungo la linea.

Due treni non possono viaggiare a distanza inferiore ai 10 km **sulle tratte prive di limiti di velocità. Tale vincolo spaziale non si applica nelle zone in cui è presente il limite di velocità di 80 km/h.**

È possibile che un treno accumuli ritardo a causa del traffico. Tale ritardo deve essere registrato dal treno.

Orario treni

L'orario indica quali treni sono in partenza dalla prima stazione verso l'ultima e quali dall'ultima verso la prima. L'orario è letto dal file **timetables.txt**, scritto secondo il seguente formato:

[numero treno (intero)]

[campo == 0 per partenza dalla stazione origine, == 1 per partenza dalla stazione capolinea]

[Tipologia di treno: == 1 per regionale, == 2 per alta velocità, ==3 per alta velocità super]

[orario di partenza] [orario arrivo stazione 1] [orario arrivo stazione 2] ... [orario arrivo stazione capolinea]

(nota: tutti i valori sono sulla stessa riga nel file di testo – gli a capo in questo documento servono solo per chiarezza).

Gli orari sono espressi con un intero che rappresenta il numero di minuti dalle ore 00:00 – il minuto rappresenta la granularità del tempo. Se il file contiene tempi di arrivo non compatibili con i limiti di velocità dei treni e delle stazioni indicati sopra, la tabella orari deve essere modificata con i tempi di percorrenza più brevi compatibili con tali limiti. **Questo significa che la tabella usata dal sistema deve considerare questi tempi modificati come quelli corretti, senza segnalare alcun ritardo se sono rispettati. Il file timetables.txt non deve essere modificato. Gli orari di arrivo alle varie stazioni considerano solo le stazioni in cui il treno deve fermarsi. Eventuali dati in eccesso devono essere**

ignorati, mentre se il numero di dati è in difetto il sistema deve completare la tabella con i tempi minimi previsti + un margine di 10 minuti.

Il sistema deve considerare un solo giorno di funzionamento. Un treno può arrivare dopo la mezzanotte di fine giornata – in questo caso, il sistema deve considerare anche i viaggi che proseguono oltre la giornata corrente, ma devono essere ignorati i treni in partenza il giorno dopo. Si supponga che nel file non esistano due treni con lo stesso numero.

Output

Il sistema deve dare un output sullo standard output a fronte dei seguenti eventi:

- Segnalazione di un treno alla stazione
- Segnalazione di una stazione al treno
- Arrivo di un treno in stazione per salita/discesa, con l'indicazione dell'orario
- Partenza di un treno dalla stazione (inclusa la prima), con l'indicazione dell'orario
- Modifica ritardo (cioè il ritardo è aumentato o diminuito rispetto a quanto segnalato prima, da segnalare in arrivo alle stazioni)
- Ritardo di arrivo di un treno alla destinazione finale
- Timetable modificata perché non compatibile con le caratteristiche dei treni

Indicazioni per lo svolgimento

Il progetto deve essere suddiviso in più file sorgente. **Ogni file deve essere scritto da un solo studente.** È tuttavia possibile controllare che il codice dei propri compagni di gruppo funzioni correttamente. Un errore in un file che inficia il funzionamento del progetto potrebbe causare una penalizzazione della valutazione dell'intero gruppo. **Il nome dell'autore deve essere indicato in un commento a inizio file.** Ovviamente, è necessario e positivo che si discuta all'interno di ciascun gruppo su come realizzare il codice, ma ogni studente è responsabile della gestione del codice che deve scrivere.

I nomi di file che sono espressamente citati in questo documento devono essere utilizzati **obbligatoriamente** (inclusa la distinzione maiuscole/minuscole).

Saranno valutati:

- Chiarezza e correttezza del codice;
- Corretta gestione della memoria e delle strutture dati;
- Efficienza del codice e della soluzione trovata;
- Utilizzo di strumenti appropriati.

La pianificazione del traffico ferroviario è un problema potenzialmente molto complesso. Tuttavia, lo scopo di questo progetto è la creazione di software, quindi eventuali soluzioni sub-ottime non saranno penalizzate. In altre parole: il focus è sulla produzione di software più che sulla ottimalità dell'algoritmo di gestione del traffico.

Il software deve essere basato unicamente sulla libreria standard del C++.

Consegna

Il compito deve essere consegnato su Moodle da un componente del gruppo, caricando un archivio che include una sola directory chiamata Rail, contenente:

- Il codice sorgente (eventualmente organizzato in sottodirectory);
- Il file CMakeLists.txt necessario alla compilazione (uno solo e posto nella directory principale del progetto)

- Almeno due file `line_description.txt` e `line_description_1.txt` che avete testato come descrizione della linea (quello letto dal software è sempre `line_description.txt`, ma voi avete provato il contenuto di entrambi i file);
- Quattro file di orario che avete provato (`timetables.txt`, `timetables_1.txt`, ecc.);
- Un file di testo contenente qualche esempio di esecuzione;
- Un file `readme.txt` in cui riportate eventuali note che volete comunicare in fase di correzione. Questo file non è la documentazione, che deve essere inserita sotto forma di commenti nel codice, ma un elenco di note aggiuntive per chi corregge, per esempio problemi riscontrati e non risolti.

L'archivio non deve contenere l'eseguibile, perché il sorgente sarà compilato in fase di correzione. Il sistema CMake deve compilare con le opzioni di ottimizzazione attivate (-O2).

Dopo la consegna su moodle, **verificate ciò che avete consegnato** con i passi seguenti:

- Scaricate il vostro progetto da moodle in una directory diversa da quella usata per sviluppare;
- Lanciare `cmake`;
- Compilare il codice e verificare la corretta esecuzione.

Suggerisco di consegnare anche compiti non completi. È tuttavia necessario che il software consegnato sia compilabile ed eseguibile.