

Pesaro, 17 Giugno 2021



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

ALGORITMI E STRUTTURE DATI

PROGETTO SESSIONE ESTIVA A.A: 2020/2021

Giulio Maria Bianchi

matricola 301993

Secondo anno di corso

1) SPECIFICA DEL PROGETTO

Si supponga di dover progettare un programma per un sistema informativo che gestisce i dati relativi ad un'anagrafe di autoveicoli. Il sistema permette di mantenere, per ogni veicolo, questi dati:

- veicolo: codice univoco alfanumerico rappresentato da una stringa concatenazione di quattro lettere e due numeri;
- proprietario: codice rappresentato da una stringa concatenazione di 3 lettere e tre numeri;
- modello del veicolo: stringa di massimo 20 caratteri;
- anno di immatricolazione: un numero intero.

Le informazioni sono memorizzate su un file in formato testo, secondo il seguente formato (si assumano campi separati da tabulazione o spazio):

Ad esempio:

Veicolo	Proprietario	Modello	Anno
ZZYQ48	FCA001	Fiat 850	1973
CCGH07	DEF130	Fiat 500	1971
AAAB12	ABC001	Ford T	1908
...

.Si scriva un programma ANSI C che esegue le seguenti elaborazioni:

1. Acquisisce il file e memorizza le relative informazioni in una struttura dati di tipo dinamico.
2. Ricerca e restituisce i dati relativi ad un dato veicolo. Ad esempio: se l'utente chiede i dati relativi al veicolo CCGH07, il programma deve restituire le informazioni contenute nella riga corrispondente:

Veicolo	Proprietario	Modello	Anno
CCGH07	DEF130	Fiat 500	1971
3. Inserimento di dati relativi ad un nuovo veicolo, specificati dall'utente.
4. Rimozione di dati relativi ad un veicolo, specificati dall'utente.
5. Calcolo del valore più grande e di quello più piccolo (secondo l'ordine lessicografico) del codice alfanumerico rappresentante il veicolo (AAAB12 e ZZYQ48, rispettivamente, nell'esempio riportato).

Per quanto riguarda l'analisi teorica si deve fornire la complessità:

- dell'algoritmo di ricerca
- dell'algoritmo di inserimento
- dell'algoritmo di rimozione
- dell'algoritmo di calcolo del valore più grande
- dell'algoritmo di calcolo del valore più piccolo

Oltre all'analisi teorica della complessità si deve effettuare uno studio sperimentale della stessa per le operazioni sopra specificate (ricerca, inserimento, rimozione, calcolo del massimo, calcolo del minimo). Come suggerimento si può operare generando un numero N di dati-veicolo casuali (dove N rappresenta il numero di veicoli). L'analisi sperimentale deve quindi valutare la complessità al variare di N

2) ANALISI DEL PROBLEMA

INPUT:

- numero intero indicante i veicoli da inserire nel file contenente un'elenco di autoveicoli

In base alla scelta dell'utente:

- 1) se si sceglie l'opzione di ricerca, viene chiesto all'utente un veicolo da ricercare
- 2) se si sceglie l'opzione di inserimento viene chiesto all'utente il veicolo da inserire in lista
- 3) se si sceglie l'opzione di rimozione viene chiesto all'utente il veicolo da eliminare dalla

lista

OUTPUT

- in base alla scelta fatta dall'utente:
 1. nome e attributi riguardanti uno specifico veicolo ricercato
 2. esito positivo o negativo riguardo all'operazione di inserimento
 3. esito positivo o negativo riguardo all'operazione di rimozione
 4. nome del veicolo più grande (secondo l'ordine lessicografico) del codice alfanumerico
 5. nome del veicolo più piccolo (secondo l'ordine lessicografico) del codice alfanumerico
- Indipendentemente dalla scelta dell'utente vengono comunicati il numero dei passi che svolge rispettivamente:
 1. l'algoritmo di ricerca
 2. l'algoritmo di inserimento
 3. l'algoritmo di rimozione
 4. l'algoritmo di ricerca del massimo
 5. l'algoritmo di ricerca del minimo

RELAZIONI:

- Dato un numero intero inserito dall'utente, vengono generati un corrispondente numero di nodi di tipo "veicolo" che vengono inseriti all'interno di un file.
- Tale file sarà dato in pasto al programma.
- In base al carattere inserito dall'utente nel menù di scelta, viene eseguita la ricerca del veicolo o l'inserimento di un veicolo o la rimozione del campo veicolo o la ricerca del minimo lessicografico o massimo lessicografico.

3) PROGETTAZIONE DELL'ALGORITMO

- Acquisire il numero di veicoli che si intende manipolare tramite le funzioni. Questo sarà utile anche per facilitare le operazioni relative al calcolo sperimentale della complessità.
- Creazione in automatico di un file con veicoli pseudocasuali che sarà acquisito dal programma
- Scelta dell'utente dell'operazione che desidera effettuare tra: ricerca, inserimento, rimozione, ricerca massimo lessicografico, minimo lessicografico.
- Conteggio del numero di passi necessari ad ogni funzione per produrre il risultato desiderato.

4) IMPLEMENTAZIONE DELL'ALGORITMO

- L'apporccio scelto è modulare, suddivendo il programma in più funzioni per rendere il codice più leggibile e velocizzare la ricerca di eventuali errori in fase di esecuzione.
- Per rappresentare un insieme di dati strutturato come un elenco anagrafico si è deciso di implementare una struttura dinamica di tipo lista lineare che permette di ordinare logicamente i dati attraverso dei nodi.
- Per rappresentare un veicolo e i suoi attributi è stato definito un nuovo tipo di dato struct chiamato «veicolo», una struttura formata da 5 campi di cui tre stringhe, uno intero e un campo puntatore all'elemento successivo.
- Tale struttura dati richiede che venga individuato il primo nodo detto "testa" il quale permette di accedere a tutti gli altri nodi della lista sequenzialmente. Inoltre ogni funzione accede alla lista grazie all'indirizzo di "testa" della lista.
- Non essendo in grado di individuare il numero di input a tempo di compilazione occorre utilizzare il meccanismo dell'allocazione dinamica che garantisce un risparmio di memoria rispetto ad altre strutture come gli array e permette di cambiare dimensioni della lista a tempo di esecuzione.
- Nelle funzioni di inserimento e rimozione il parametro formale della funzione viene passato per doppio asterisco per rendere visibili le modifiche della testa al programma chiamante main.
- Le funzioni di inserimento e rimozione restituiscono come risultato l'esito positivo o negativo dell'operazione mentre le funzioni di ricerca, ricerca del massimo lessicografico, e del minimo lessicografico restituiscono il veicolo ricercato.
- Per generare dei veicoli con caratteri alfanumerici si è deciso di utilizzare la funzione srand con la libreria «time.h».
- Per effettuare la verifica sperimentale sono stati utilizzati delle variabili che si incrementano ad ogni istruzione che svolge la specifica funzione al fine di contare i passi degli algoritmi di ricerca, inserimento, rimozione, massimo e minimo lessicografico.
- Si è preferito valutare l'efficienza in base ad un contatore rispetto alla misurazione del tempo impiegato dall'algoritmo per risolvere le varie operazioni perché prescinde dalle prestazioni del calcolatore utilizzato e dal linguaggio utilizzato.

5) TESTING DEL PROGRAMMA

Nei seguenti screenshot vengono forniti i risultati delle istruzioni necessarie per svolgere l'algoritmo di ricerca in base agli input forniti:

TEST 1: 1 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
1
Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 2
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 9.567 s
Press any key to continue.
```

TEST 2: 10 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
10
Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 22
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 6.799 s
Press any key to continue.
```

TEST 3: 50 INPUT

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
50

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
bbb
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 102
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 4.512 s
Press any key to continue.
```

TEST 4: 100 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
100

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 202
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 4.954 s
Press any key to continue.
```

TEST 5: 200 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi\C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
200

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
zzz
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 402
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 5.165 s
Press any key to continue.
```

TEST 6: 500 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi\C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
500

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
z
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 1002
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 3.822 s
Press any key to continue.
```


TEST 7: 1000 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\C\Esercizi C\Nuovo\quasi definitivo.exe"
quanti veicoli vuoi creare?
1000

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento dati di un veicolo*****
'3' *****Rimozione dati di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
'6' *****Esci dal programma*****
1
Che veicolo vuoi cercare?
d
elemento non trovato

Il numero dei passi dell'algoritmo di inserimento sono 0
Il numero dei passi dell'algoritmo di rimozione sono 0
Il numero dei passi dell'algoritmo di ricerca sono 2002
Il numero dei passi dell'algoritmo del massimo in ordine lessicografico sono 0
Il numero dei passi dell'algoritmo del minimo in ordine lessicografico sono 0

Process returned 0 (0x0)   execution time : 3.939 s
Press any key to continue.
```

TEST 8: 2000

```
"C:\Users\Giulio Maria Bianchi\Desktop\quasi definitivo.exe"
Inserire il numero di veicoli da generare in modo pseudocasuale...
*****i veicoli generati confluiranno all'interno del file di input che verra' acquisito dal programma*****
2000

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento di un veicolo e relativi attributi*****
'3' *****Rimozione di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero di passi dell'algoritmo di inserimento e' 0
Il numero di passi dell'algoritmo di rimozione e' 0
Il numero di passi dell'algoritmo di ricerca e' 4001
Il numero di passi dell'algoritmo del massimo in ordine lessicografico e' 0
Il numero di passi dell'algoritmo del minimo in ordine lessicografico e' 0

Process returned 0 (0x0)   execution time : 7.985 s
Press any key to continue.
```

TEST 9 : 10000 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\quasi definitivo.exe"
Inserire il numero di veicoli da generare in modo pseudocasuale...
*****i veicoli generati confluiranno all'interno del file di input che verra' acquisito dal programma*****
10000

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento di un veicolo e relativi attributi*****
'3' *****Rimozione di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero di passi dell'algoritmo di inserimento e' 0
Il numero di passi dell'algoritmo di rimozione e' 0
Il numero di passi dell'algoritmo di ricerca e' 20001
Il numero di passi dell'algoritmo del massimo in ordine lessicografico e' 0
Il numero di passi dell'algoritmo del minimo in ordine lessicografico e' 0

Process returned 0 (0x0)   execution time : 16.901 s
Press any key to continue.
```

TEST 10 : 50000 input

```
"C:\Users\Giulio Maria Bianchi\Desktop\quasi definitivo.exe"
Inserire il numero di veicoli da generare in modo pseudocasuale...
*****i veicoli generati confluiranno all'interno del file di input che verra' acquisito dal programma*****
50000

Inserire un numero:
'1' *****Ricerca un veicolo*****
'2' *****Inserimento di un veicolo e relativi attributi*****
'3' *****Rimozione di un veicolo*****
'4' *****Valore piu' grande in ordine lessicografico*****
'5' *****Valore piu' piccolo in ordine lessicografico*****
1
Che veicolo vuoi cercare?
aaa
elemento non trovato

Il numero di passi dell'algoritmo di inserimento e' 0
Il numero di passi dell'algoritmo di rimozione e' 0
Il numero di passi dell'algoritmo di ricerca e' 100001
Il numero di passi dell'algoritmo del massimo in ordine lessicografico e' 0
Il numero di passi dell'algoritmo del minimo in ordine lessicografico e' 0

Process returned 0 (0x0)   execution time : 35.911 s
Press any key to continue.
```

6) VALUTAZIONE DELLA COMPLESSITA' DEL PROGRAMMA

1) ALGORITMO DI RICERCA

- Nel caso ottimo in cui il valore da ricercare è il primo della lista la complessità è costante:

$$T(n) = 1 + 1 = 2 = \mathbf{O(1)}$$

- Nel caso pessimo in cui il valore da ricercare non è presente all'interno della lista la complessità è lineare:

$$T(n) = 1 + n * (1 + 1) + 1 = 2n + 2 = \mathbf{O(n)}$$

2) ALGORITMO DI INSERIMENTO

- Nel caso ottimo in cui il valore da inserire è in prima posizione la complessità è costante:

$$T(n) = T(n) = 1 + 1 + 1 + 1 = 4 = \mathbf{O(1)}$$

- Nel caso pessimo in cui l'elemento da inserire è in ultima posizione, la complessità è lineare:

$$T(n) = 1 + n(1 + 1) + 1 + 10 = 2n + 12 = \mathbf{O(n)}$$

3) ALGORITMO DI RIMOZIONE

- Nel caso ottimo in cui l'elemento da rimuovere è la testa, la complessità è costante:

$$T(n) = 1 + 1 + 1 + 1 = 4 = \mathbf{O(1)}$$

- Nel caso pessimo in cui l'elemento da rimuovere non è presente all'interno della lista, la complessità è lineare:

$$T(n) = 1 + n(1 + 1) + 1 + 10 = 2n + 12 = \mathbf{O(n)}$$

4) ALGORITMI DI CALCOLO DEL MINIMO/MASSIMO IN ORDINE LESSICOGRAFICO:

- La trattazione della complessità dei due algoritmi viene equiparata in quanto ambedue hanno la stessa struttura implementativa (hanno costrutti identici).

- Nel caso ottimo, in cui il primo elemento della lista è il minimo/massimo, è sufficiente fermarsi al primo elemento e la complessità è costante:

$$T(n) = 1 + 1 + (1 + 1) + 1 = \mathbf{O(1)}$$

- Nel caso pessimo, in cui l'elemento non è presente in lista, occorre scorrere tutta la lista e la complessità è lineare:

$$T(n) = 1 + 1 + n(1 + 1) + 1 = 2n + 3 = \mathbf{O(n)}$$

I casi medi di ogni algoritmo non sono stati trattati a lezione in quanto fanno affidamento sulla media ponderata delle diverse istanze di ingresso.

Viene fornito un grafico indicativo dell'andamento della curva asintotica dell'algoritmo di ricerca nel caso pessimo in cui l'elemento che si ricerca non è presente all'interno della lista. Il grafico, al netto di costanti, risulta essere pressochè identico a quello degli algoritmi di inserimento, rimozione, ricerca del massimo lessicografico e del minimo lessicografico nel caso pessimo.

