

Esame 20220111

Esercizio 3

(1) Esercizio 3 v1

ESSAY marked out of 10 penalty 0 File picker

Scrivere nel file `esercizio3.cc` un programma che riceve da linea di comando (quindi usando `argc` e `argv`) il percorso di un file contenente un numero intero per riga. Il file contiene almeno 1 numero intero e al massimo 100 interi, e tutti gli interi sono diversi fra loro. Il programma deve creare un albero di ricerca binaria **bilanciato** con i numeri contenuti nel file.

Questi sono due esempi di esecuzione:

<pre>computer > cat input.txt 1 3 2 4 5 7 6 8 computer > ./a.out input.txt Stampa BST bilanciato: 8 7 6 5 4 3 2 1</pre>		<pre>computer > cat input.txt 8 2 5 7 1 4 3 6 computer > ./a.out input.txt Stampa BST bilanciato: 8 7 6 5 4 3 2 1</pre>
--	--	--

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per implementare le funzionalità richieste e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di deallocare la memoria allocata dinamicamente;
- Implementare controlli sulla buona riuscita delle operazioni (e.g., l'apertura del file di input) e stampare a video messaggi d'errore appropriati;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `fstream`.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

Suggerimenti:

- Ricordarsi che l'albero di ricerca binaria dev'essere **bilanciato**. Quindi, il primo elemento contenuto nel file di input non dev'essere necessariamente la radice dell'albero. Gli elementi contenuti nel file di input non sono necessariamente ordinati.
- Il file `esercizio3.cc` contiene già una funzione `stampaAlbero` che stampa a terminale l'albero evidenziando le relazioni fra i nodi. Potete utilizzare la funzione `stampaAlbero` per controllare se l'albero è bilanciato.

Information for graders:

(2) Esercizio 3 v2

ESSAY

marked out of 10

penalty 0

File picker

Scrivere nel file `esercizio3.cc` un programma che riceve da linea di comando (quindi usando `argc` e `argv`) il percorso di un file contenente una lettera dell'alfabeto per riga (26 lettere). Il file contiene almeno 1 lettera e al massimo 26 lettere, e tutte le lettere sono diverse fra loro. Il programma deve creare un albero di ricerca binaria **bilanciato** con le lettere contenute nel file, ordinate alfabeticamente.

Questi sono due esempi di esecuzione:

<pre>computer > cat input.txt a c b d e g f h computer > ./a.out input.txt Stampa BST bilanciato: h g f e d c b a</pre>		<pre>computer > cat input.txt h b e g a d c f computer > ./a.out input.txt Stampa BST bilanciato: h g f e d c b a</pre>
---	--	---

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per implementare le funzionalità richieste e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di deallocare la memoria allocata dinamicamente;
- Implementare controlli sulla buona riuscita delle operazioni (e.g., l'apertura del file di input) e stampare a video messaggi d'errore appropriati;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `fstream`.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

Suggerimenti:

- Ricordarsi che l'albero di ricerca binaria dev'essere **bilanciato**. Quindi, il primo elemento contenuto nel file di input non dev'essere necessariamente la radice dell'albero. Gli elementi contenuti nel file di input non sono necessariamente ordinati.

- Il file `esercizio3.cc` contiene già una funzione `stampaAlbero` che stampa a terminale l'albero evidenziando le relazioni fra i nodi. Potete utilizzare la funzione `stampaAlbero` per controllare se l'albero è bilanciato.

Information for graders:

(3) Esercizio 3 v3

ESSAY

marked out of 10

penalty 0

File picker

Scrivere nel file `esercizio3.cc` un programma che riceve da console input (quindi da `cin`) il percorso di un file (il percorso è lungo al massimo 30 caratteri). Il file contiene un numero intero per riga. Il file contiene almeno 1 numero intero e al massimo 100 interi, e tutti gli interi sono diversi fra loro. Il programma deve creare un albero di ricerca binaria **bilanciato** con i numeri contenuti nel file.

Questi sono due esempi di esecuzione:

<pre>computer > cat input.txt 1 3 2 4 5 7 6 8 computer > ./a.out Inserisci il file: input.txt Stampa BST bilanciato:</pre>		<pre>computer > cat input.txt 8 2 5 7 1 4 3 6 computer > ./a.out Inserisci il file: input.txt Stampa BST bilanciato:</pre>
<pre> 8</pre>		<pre> 8</pre>
<pre> 7</pre>		<pre> 7</pre>
<pre> 6</pre>		<pre> 6</pre>
<pre> 5</pre>		<pre> 5</pre>
<pre>4</pre>		<pre>4</pre>
<pre> 3</pre>		<pre> 3</pre>
<pre> 2</pre>		<pre> 2</pre>
<pre> 1</pre>		<pre> 1</pre>

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per implementare le funzionalità richieste e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di deallocare la memoria allocata dinamicamente;
- Implementare controlli sulla buona riuscita delle operazioni (e.g., l'apertura del file di input) e stampare a video messaggi d'errore appropriati;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

Suggerimenti:

- Ricordarsi che l'albero di ricerca binaria dev'essere **bilanciato**. Quindi, il primo elemento contenuto nel file di input non dev'essere necessariamente la radice dell'albero. Gli elementi contenuti nel file di input non sono necessariamente ordinati.
- Il file `esercizio3.cc` contiene già una funzione `stampaAlbero` che stampa a terminale l'albero evidenziando le relazioni fra i nodi. Potete utilizzare la funzione `stampaAlbero` per controllare se l'albero è bilanciato.

Information for graders:

(4) Esercizio 3 v4

ESSAY

marked out of 10

penalty 0

File picker

Scrivere nel file `esercizio3.cc` un programma che riceve da console input (quindi da `cin`) il percorso di un file (il percorso è lungo al massimo 30 caratteri). Il file contiene una lettera dell'alfabeto per riga (26 lettere). Il file contiene almeno 1 lettera e al massimo 26 lettere, e tutte le lettere sono diverse fra loro. Il programma deve creare un albero di ricerca binaria **bilanciato** con le lettere contenute nel file, ordinate alfabeticamente.

Questi sono due esempi di esecuzione:

<pre>computer > cat input.txt a c b d e g f h computer > ./a.out Inserisci il file: input.txt Stampa BST bilanciato:</pre>		<pre>computer > cat input.txt h b e g a d c f computer > ./a.out Inserisci il file: input.txt Stampa BST bilanciato:</pre>
<pre> h / \ g \ / \ \ f e \ / \ \ \ d c \ \ / \ \ \ \ b a \ \ \</pre>		<pre> h / \ g \ / \ \ f e \ / \ \ \ d c \ \ / \ \ \ \ b a \ \ \</pre>

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per implementare le funzionalità richieste e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di deallocare la memoria allocata dinamicamente;
- Implementare controlli sulla buona riuscita delle operazioni (e.g., l'apertura del file di input) e stampare a video messaggi d'errore appropriati;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

Suggerimenti:

- Ricordarsi che l'albero di ricerca binaria dev'essere **bilanciato**. Quindi, il primo elemento contenuto nel file di input non dev'essere necessariamente la radice dell'albero. Gli elementi contenuti nel file di input non sono necessariamente ordinati.
- Il file `esercizio3.cc` contiene già una funzione `stampaAlbero` che stampa a terminale l'albero evidenziando le relazioni fra i nodi. Potete utilizzare la funzione `stampaAlbero` per controllare se l'albero è bilanciato.

Information for graders:

Total of marks: 40