

Esame di Algoritmi e Laboratorio - Modulo di Laboratorio

Appello del 17 luglio 2020

Note

Completata la prova, spedire il codice sorgente, costituito da un singolo file chiamato “main.cpp”, all’indirizzo `santamaria@dmi.unict.it`. Prove inviate oltre l’orario concordato non saranno valutate. Indicare nell’oggetto del messaggio “Esame Laboratorio di Algoritmi”, nel corpo del messaggio nome, cognome, matricola e compilatore utilizzato.

Specifiche

Si implementi una coda di minima priorità attraverso la struttura dati di tipo Heap. La struttura dati deve contenere, tra gli altri, i seguenti tre metodi:

- *enqueue*. Questo metodo consente di inserire un valore nella struttura dati mantenendo valide tutte le sue proprietà.
- *modifyKey*. Questo metodo consente di sostituire il valore di una chiave con un altro, noto l’indice della posizione nella struttura dati. È possibile sostituire il valore di una chiave sia con un valore minore che con uno maggiore del valore della chiave stessa, mantenendo valide le proprietà della Heap.
- *isHeap*. Questo metodo restituisce un valore booleano che indica se la struttura dati è una Heap valida.

Si testi la struttura dati con l’input fornito.

L’input è suddiviso in task, uno per ogni riga. Ogni riga del file di input è formata da $3+m+n$ elementi. Il primo elemento è una stringa che indica il tipo di dato da inserire (int o double). Il secondo è un intero che rappresenta il numero di elementi da inserire attraverso un’operazione di enqueue. Il terzo è il numero di operazioni di modifyKey che devono essere effettuate. Seguono gli m valori da inserire nella struttura dati attraverso un’operazione di enqueue e le n operazioni di modifyKey. Ciascuna operazione è nel formato $(i\ k)$ (con le parentesi tonde e lo spazio nel mezzo), dove i è la posizione della chiave da modificare e k è il valore che deve sostituire il valore della chiave in posizione i .

L'output dovrà essere formato da una riga per ogni task. Ogni riga contiene gli m valori della struttura dati dell' i -esimo task, separati da uno spazio.

Esempio

Il seguente esempio presenta un file di input contenente 3 linee di input ed il corrispondente file di output (vedere i file allegati).

```
int 19 8 92 29 95 6 14 27 29 28 92 39 85 64 2 73 21 11 82 63 8 (7 33) (11 28)
(10 83) (16 99) (14 0) (12 79) (7 55) (4 9)
```

```
double 9 9 3.5 7.1 8.5 2.0 9.1 7.5 0.7 6.9 9.6 (9 1.8) (3 7.3) (6 5.7) (2 3.3) (7 0.7)
(5 4.3) (3 6.9) (1 0.0) (6 0.5)
```

```
double 4 4 8.0 2.7 8.1 1.7 (3 4.6) (2 4.1) (4 0.4) (2 6.2)
```

Il corrispondente file di output sarà:

```
0 8 2 9 28 27 29 28 14 83 29 79 64 55 33 99 82 92 63
0.0 3.3 0.5 3.5 4.3 5.7 6.9 7.1 6.9
0.4 6.2 4.6 4.1
```