

DIRETTIVE

Il progetto è composto da 6 test da 3 subtask ciascuno.

Il subtask 1 vale 3 punti, i restanti 1 punto.

Per raggiungere la sufficienza l'ultima sottoposizione deve passare almeno il primo subtask di ogni test.

I nomi dei test e le specifiche qui sotto (tutorial vale 0 e Lode vale 1 ottenibile solo con 30)

| Problema | Nome | Limite di tempo | Limite di memoria |
|------------------------|-----------------|-----------------|-------------------|
| Tutorial | Tutorial | 1,000 secondo | 256 MiB |
| WriteOnly | WriteOnly | 7,400 secondi | 3,00 GiB |
| BulkReads | BulkReads | 1,250 secondi | 240 MiB |
| TimeForAChange | TimeForAChange | 2,100 secondi | 540 MiB |
| RollingBack | RollingBack | 2,100 secondi | 550 MiB |
| AlteringHistory | AlteringHistory | 2,100 secondi | 400 MiB |
| Rollercoaster | Rollercoaster | 2,700 secondi | 1,03 GiB |
| Lode | Lode | 2,000 secondi | 340 MiB |

I primi 3 test provano le funzioni print, delete e change. Gli ultimi 3 test tutte le operazioni.

Versione prog1 (scritte le funzioni change, delete e print)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Versione prog2 (scritte le funzioni change, delete, print e undo e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 0/5 |
| | 2 | punteggio 0/5 |
| | 3 | punteggio 0/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

update 29/8: scoperto errore. Sottoponendo il codice decommentando pezzo a pezzo tutto quello diverso da prog1 ho scoperto che fp_list, un semplice puntatore di interi, per qualche motivo che il mio compilatore non vede fa andare in segmentation fault l'esecuzione del programma. Ad ogni modo, tenendo commentato quella parte di codice e aumentando la memoria molto lentamente per star all'interno dei parametri previsti, il codice che mi va a salvare il testo comunque occupa troppo tempo uscendo dai parametri.

Prog2 è un fallimento totale come progettazione. Ripartirò con prog1, salvando il salvabile da prog2 e ripenserò ad undo e redo.

Versione prog3 (scritte le funzioni change, delete, print e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 4/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 30/08: La struttura ora implementata spreca molto meno spazio di quella precedente ottenendo buoni risultati nei primi 3 test. Tuttavia con questa struttura non riesco a pensare ad una undo e redo che abbiano complessità temporale costante. Secondo le faq del progetto per passare correttamente i risultati successivi le operazioni di undo e redo devono essere indipendenti dal numero di righe scritte (cosa che faceva prog2 ma senza una struttura sufficientemente ottimizzata).

Ripartendo da prog3 proverò ad ottimizzare la struttura dati in modo tale che l'undo e redo non siano un taglia/incolla sul mio testo principale. Vorrei arrivare alla fine non avendo più un testo principale ma un vettore di testi con una struttura parallela che mi dica sempre come stampare e l'undo e la redo non siano altro che una operazione sugli indici. Per fare ciò sarà necessario cambiare la funzione print.

Versione prog4 (scritte le funzioni change, delete, print e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 31/8: La struttura implementata ora è completamente diversa e spreca uno spazio che dipende solo dal numero di righe totali date in input e da puntatori qui e là. Non sono riuscito a liberarmi del testo principale, tuttavia se prima vi erano diverse copia incolla delle stringhe in varie strutture, ora esiste un unico elenco di stringhe e tutte le altre strutture sono un elenco di puntatori a quelle stringhe. L'undo e la redo così facendo però non possono essere a tempo costante, attualmente non ho in mente una soluzione migliore. Cercherò di ottimizzare i tempi accorpendo undo e redo.

Versione prog5 (scritte le funzioni change, delete, print e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 2/9: Da abbandonare. Letteralmente impossibile scrivere la funzione inversa della change. Prog6 dovrà essere una via di mezzo tra prog5 e prog2, dovrà avere l'utilizzo della prog2 ma lo spreco di memoria di prog5, sono fiducioso.

Versione prog6 (scritte le funzioni change, delete, print, undo e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 0/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 3/9: Forse è la volta buona. L'undo sembra funzionare correttamente e ma l'utilizzo della memoria è eccessivo. WriteOnly sfiora i 3 GB. Permangono dei bug da correggere ma sono fiducioso.

Versione prog7 (scritte le funzioni change, delete, print, undo e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 0/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 4/9: Ora la struttura si fa ben articolata. Ho trovato un modo efficace di ottimizzare la memoria. Nel caso in cui la modifica del testo vada solo ad aggiungere frasi e non a sovrascriverne, non copio tutto il testo da allocare nello storico dei testi, bensì ne copio solo la parte finale. Così facendo diminuisco di molto lo spazio occupato nel test WriteOnly, passando dai 3GB a 40MB. Domani si scrive al redo!

Versione prog8 (scritte le funzioni change, delete, print, undo, redo e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 5/5 |
| | 5 | punteggio 0/5 |
| | 6 | punteggio 0/5 |

Update 5/9: Si vede la luce, niente da aggiungere se non che ho implementato la redo. Rimangono dei casi in cui vado in segmentation fault. Ho dei dubbi sulla riscrittura del testo e sulle free, ma ci siamo.

Versione 10613915_ZENI_WILLIAM (scritte le funzioni change, delete, print, undo, redo e implementata struttura undo)

| | | |
|-------------|---|----------------------|
| Test | 1 | punteggio 5/5 |
| | 2 | punteggio 5/5 |
| | 3 | punteggio 5/5 |
| | 4 | punteggio 5/5 |
| | 5 | punteggio 5/5 |
| | 6 | punteggio 5/5 |
| | 7 | punteggio 1/1 |

Update 6/9: Finalmente concluso. Corretti gli ultimi bug ed eliminate tutte le free. Il codice non è agevole e poteva essere sicuramente migliorato ma passa tutti i test con il max dei punteggi e stando di molto sotto le specifiche. Sono soddisfatto.
Voto Simulato: 30 Lode