

Laboratorio di algoritmi e strutture dati

Ancora sugli alberi*

Docente: Violetta Lonati

In questa esercitazione si deve scrivere/completare il programma `foresta.go` definendo alcune funzioni di segnatura assegnata. Potete collaudare il vostro programma usando i file di test allegati (si legga in fondo come eseguire i test).

Foresta di oggetti

Federico, appassionato di enigmi, ha un negozio di oggettistica e arredamento di seconda mano, e ha un modo tutto particolare di indicare i prezzi degli oggetti in vendita. Ad ogni oggetto, associa un indizio che serve per stabilire il prezzo dell'oggetto.

L'indizio associato ad un oggetto può essere di due tipi:

- un numero,
- oppure una scritta di questo tipo: “ $a \text{ OP } b$ ”, dove a e b sono nomi di altri oggetti e OP è il simbolo di una operazione aritmetica (+, -, *, /).

Usando gli indizi, si possono calcolare i prezzi degli oggetti:

- se l'indizio associato all'oggetto x è un numero, allora quello è il prezzo dell'oggetto x ;
- se l'indizio associato all'oggetto x è dato da un'operazione “ $a \text{ OP } b$ ”, allora il prezzo di x si ottiene combinando con l'operazione OP i prezzi dei due oggetti a e b .

Ogni oggetto compare al massimo in un indizio associato ad altri oggetti.

Esempio. Ecco gli indizi che ha predisposto Federico:

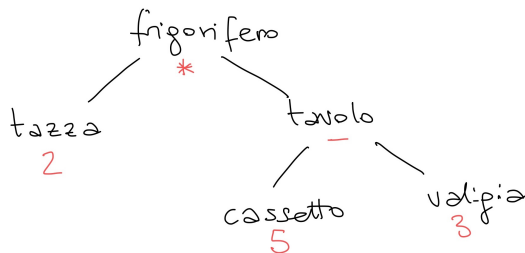
- al tavolo ha associato l'indizio `cassetto - valigia`
- al frigorifero ha associato l'indizio `tazza * tavolo`
- alla tazza ha associato l'indizio `2`
- alla valigia ha associato l'indizio `3`
- al cassetto ha associato l'indizio `5`,

Calcoliamo il prezzo del frigorifero, cui è associato l'indizio `tazza * tavolo`. Alla tazza e al tavolo sono associati rispettivamente gli indizi `2` e `cassetto - valigia`. Al cassetto e alla valigia sono associati rispettivamente gli indizi `5` e `3`. La differenza tra `5` e `3` è `2`, e questo è il prezzo del tavolo. Dunque il prezzo frigorifero è $2 * 2 = 4$.

*Ultimo aggiornamento: 15 novembre 2023 - 23:36:33

Modellazione. Modelliamo il problema usando una foresta di alberi binari. Ogni oggetto è rappresentato da un nodo; ogni nodo ha per chiave una stringa: il nome dell'oggetto che il nodo rappresenta. Gli oggetti che hanno per indizio un numero saranno le foglie; gli oggetti che non compaiono in nessun indizio sono radici; gli oggetti che hanno per indizio un'operazione saranno nodi interni. Se un nodo rappresenta un oggetto che ha per indizio l'operazione “ a OP b ”, allora il figlio sinistro del nodo ha per chiave a , il figlio destro ha per chiave b .

Es: gli indizi elencati nell'elenco precedente possono essere modellati con una foresta di un solo albero, come nel disegno qui sotto:



Completate il programma foresta.go

1. Scrivete una funzione con segnatura
`leggiInput() map[string]*oggetto`
 che legga da standard input gli indizi (uno per riga, nel formato “oggetto: indizio”), crei un oggetto per ogni indizio e li memorizzi in una mappa.
2. Definite opportunamente un tipo `foresta` (potete definire tipi ausiliari, se lo ritenete opportuno) e scrivete una funzione con segnatura
`costruisciForesta(mappa map[string]*oggetto) foresta`
 che costruisca una foresta a partire dalla mappa degli indizi.
3. Scrivete una funzione con segnatura
`up(f foresta, n string) (string, bool)`
 che stampi la chiave del padre del nodo di chiave n . Scrivete funzioni simili di nome `dx` e `sx` che restituiscano, rispettivamente, la chiave del figlio destro e sinistro del nodo di chiave n . Se tali figli/padre non esistono, la funzione deve restituire `false` come secondo argomento.
4. Scrivete una funzione con segnatura
`stampaAlbero(f foresta, n string)`
 che stampi, in ordine simmetrico, i nodi dell'albero della foresta che ha per radice il nodo di chiave n . Nel caso delle foglie, si deve stampare tra parentesi anche il numero associato. Ad esempio, se la foresta è costituita solo dall'albero disegnato sopra e n è `frigorifero`, la funzione deve stampare:

```
tazza (val = 2)
frigorifero
```

```
cassetto (val = 5)
tavolo
valigia (val = 3)
```

Se invece `n` è `tavolo`, la funzione deve stampare:

```
cassetto (val = 5)
tavolo
valigia (val = 3)
```

Se non esiste nodo con chiave `n`, la funzione deve stampare il messaggio “non esiste `n`”.

5. Scrivete una funzione con segnatura
`calcolaPrezzo(f foresta, n string) int`
che calcoli il prezzo dell’oggetto di nome `n`.

Collaudate il vostro programma usando i file nell’archivio `foresta-test.zip`.

Per eseguire i test è necessario:

- entrare nella directory dell’esercizio
- salvare nella directory il file col <nome> indicato
- creare l’e eseguibile (`go build <nome>.go`)
- lanciare il comando `go mod init <nome>`
- lanciare il comando `go test -v` (e osservare l’output)