

Maps Recursion Bufio Rand

Docente: Dragan Ahmetovic
 mail: dragan.ahmetovic@unimi.it
 Ricevimento: su appuntamento

Tutor: Alexandru David

Sito del corso: <http://dragan.ahmetovic.it/?p=teaching>



dragan.ahmetovic.it/r.php
 (indirizzo anche sul sito)

D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

0. comprensione del codice

```
func f(a, b string) bool {
    var m = make(map[rune]rune)
    for _, r := range a {
        m[r] = r
    }
    var br = []rune(b)
    for i := 0; i < len(b); i++ {
        if m[br[i]] != 0 {
            return true
        }
    }
    return false
}
```

D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

1. count_words2.go

Create un programma che chieda una frase da stdin (usando bufio) e calcoli la frequenza delle parole della frase utilizzando una map. se la frase fornita è vuota il programma termina

```
■ Enter text:
■ i'm blue da ba di da ba da
■ map[ba:2 blue:1 da:3 di:1 i'm:1]
■ Enter text:
■
■ Byebye!
```

D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

2. anagram.go

Data una parola, creare un suo anagramma casuale

```
■ Insert a word:
■ ASDFGHJKL
■ Anagram: AFSDHGJLK
```

D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

3. anagrams.go

Data una parola, stampare tutti i suoi anagrammi

```
■ Insert a word:
■ asd
■ Anagrams:
■ asd
■ ads
■ sad
■ sda
■ das
■ dsa
```

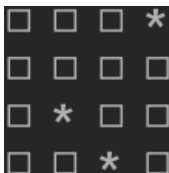
D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

4a. minesweeper1.go

Dati 3 argomenti: <altezza>, <larghezza> e <numeroBombe>, create una griglia con quelle dimensioni, e aggiungete quel numero di bombe in posizioni casuali nella griglia. Create una funzione per stampare la griglia con un '□' per ogni spazio e '*' per ogni bomba

```
■ go run main1.go 4 4 3
■ Grid:
```



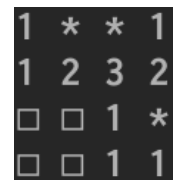
D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

4b. minesweeper2.go

Estendete l'esempio precedente per stampare, per ogni cella, il numero di bombe adiacenti (in 8 direzioni)

```
■ go run main2.go 4 4 3
■ Grid:
```



D. Ahmetovic - Maps Recursion Bufio Rand

Esercizi

4c. minesweeper3.go

completate il gioco per:

- 1 mostrare una mappa di esplorazione, inizialmente inesplorata
 - 2 permettere all'utente di esplorare una cella data la sua posizione
 - 3 se la cella è una bomba il gioco termina
 - 4 se la cella ha una o più bombe adiacenti mostrane il numero
 - 5 se la cella non ha bombe adiacenti mostra quella e tutte le celle adiacenti ad essa che non hanno bombe adiacenti (e così via)
- tip: [https://en.wikipedia.org/wiki/Flood_fill#Stack-based_recursive_implementation_\(four-way\)](https://en.wikipedia.org/wiki/Flood_fill#Stack-based_recursive_implementation_(four-way))