Teoria dos Grafos

3º Trabalho Prático

Implementação do Ford-Fulkerson
e funções auxiliares

João Sales
Matheus Cardoso

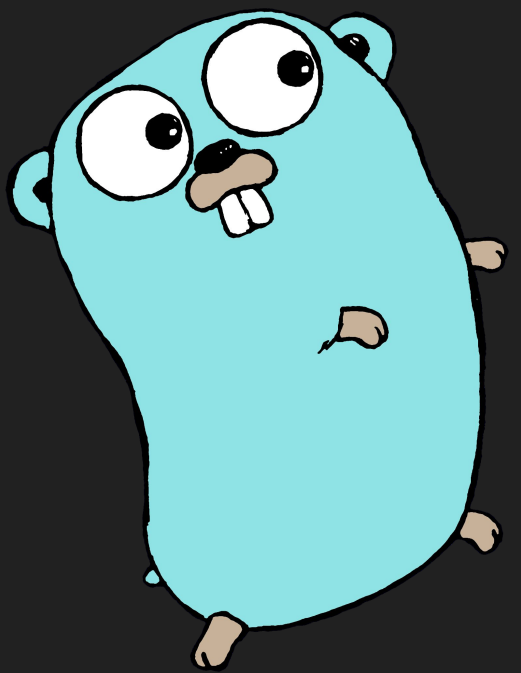
2023/2

Linguagem de Programação escolhida

Linguagem de Programação escolhida

Acesse em: **github.com/mhscardoso/goraphs**
Branch -> TP3

- Por que?

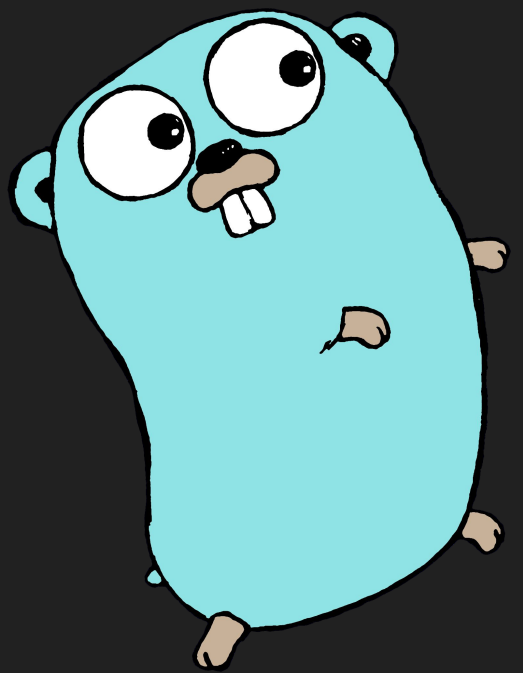


Linguagem de Programação escolhida

Acesse em: **github.com/mhscardoso/goraphs**
Branch -> TP3

- Por que?

Porque sim!!!



O mínimo necessário pra hoje:

Maps! (~= dict em Python)

```
map[int]float64
map[string]int
map[int][2]float64
map[...]...
```

O mínimo necessário pra hoje:

Para a lista de adjacências (TP3)

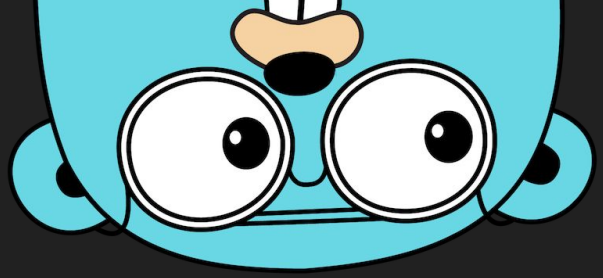
```
map[int][2]int
```

O mínimo necessário pra hoje:

Para a lista de adjacências (TP3)

```
type FList struct {
  Vector   []set.SetF
  Vertices int
  Edges    int
  Targeted bool
}
```

O mínimo necessário pra hoje:



Interfaces

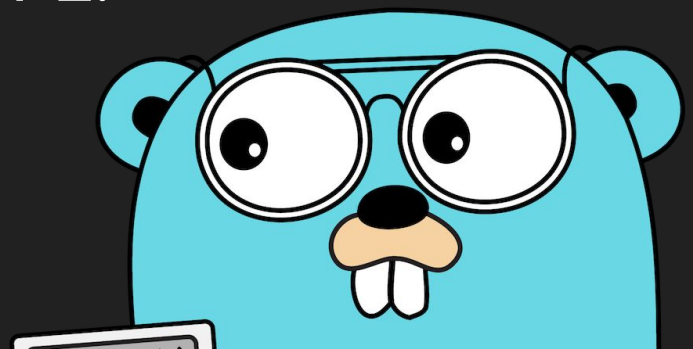
```
type Graph interface {
   Neighbors(vertex int) any
   Relate(vertex, neighbor int,
          weigth float64,
          edges *int)

   Allocate(vertices int)
   UpdateEdges(edges int)
   See()
   N() int
   M() int
}
```

Auxiliares de Ford-Fulkerson

Grafo Residual

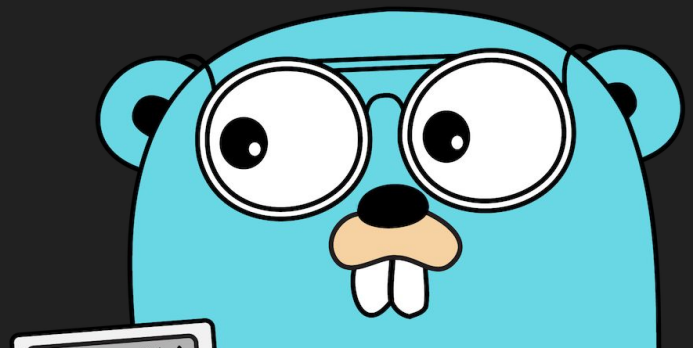
```
type WList[T float64 | int] struct {
   Vector   []set.SetW[int, T]
   Vertices int
   Edges    int
   Targeted bool
}
```

(quase) a mesma lista de adjacências
implementada no TP2!



Auxiliares de Ford-Fulkerson

```
func Bottleneck(res *awlists.WList[int],
                P []int) int
{
   if len(P) == 1 {
      return 0
   }

   b := res.Vector[P[1]][P[0]]

   for i := 2; i < len(P); i++ {
      b = min(b, res.Vector[P[i]][P[i-1]])
   }

   return b
}
```

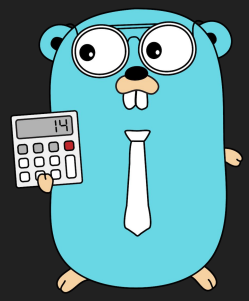


Auxiliares de Ford-Fulkerson

…show me the code!

O que importa!

Resultados:



Tempo Médio de Execução (s)	Fluxos Máximos	Deltas
0.241303	1058	2048
0.561515	11189	2048
1.925920	2964	2048
30.847329	13486	2048
69.881233	26360	16384
100.001918	26812	16384

Ford-Fulkerson

```
E se… delta == 1?
```

Ford-Fulkerson

Vamos ao Algoritmo implementado…

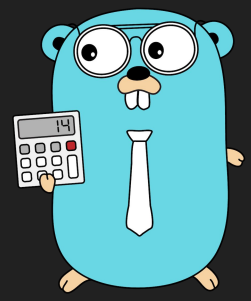
Ford-Fulkerson

Vamos ao Algoritmo implementado…

Se usarmos Delta = 1, é a mesma coisa que implementar o algoritmo sem a melhoria!

O que importa!

Resultados:



Tempo Médio de Execução (s) [delta = 1]	Tempo Médio de Execução (s)	Fluxos Máximos	Deltas
0.412467	0.241303	1058	2048
1.121258	0.561515	11189	2048
1.937627	1.925920	2964	2048
55.151556	30.847329	13486	2048
108.249299	69.881233	26360	16384
61.530668	100.001918	26812	16384

Isso é tudo, pessoal!