

ALGORITMO DE BUSCA

A^*

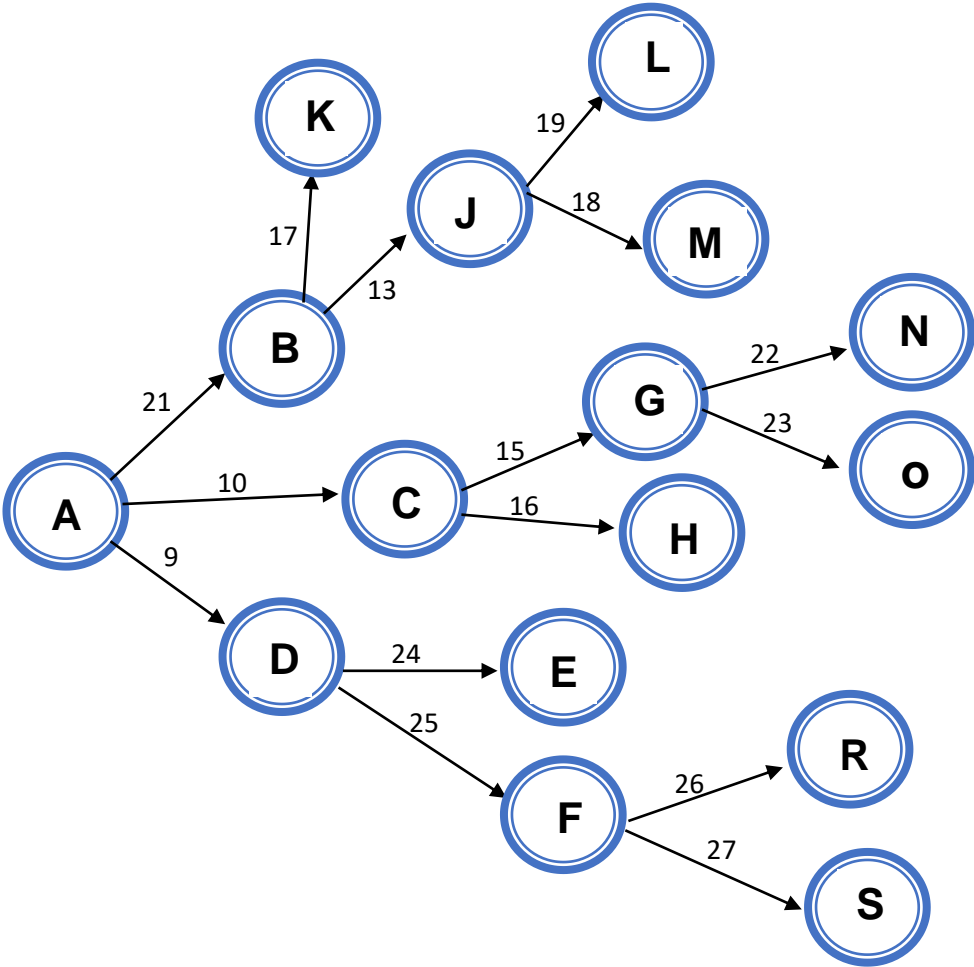
Inteligência Computacional I

Ana Cláudia Gomes Souza

Rio de Janeiro, Maio 2019

Para a implementação do algoritmo de busca A* foi utilizada a linguagem Python. A estrutura do código é composta por um TAD de cinco arquivos. Com o auxílio de um mapa composto de letras do alfabeto será realizada a entrada de dados. O objetivo do algoritmo é percorrer os estados do mapa para encontrar a letra informada pelo usuário via teclado, calcular o custo que esta busca gerou levando em consideração a distância, calcular o valor estimado e a função de avaliação.

Mapa de letras



Valores estimados

| A | B | C | D | E | F | G | H | J | K | L | M | N | O | R | S |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 11 | 20 | 39 | 14 | 35 | 10 | 26 | 33 | 27 | 29 | 28 | 32 | 13 | 36 | 37 |

Descrição dos arquivos que compõem o TAD:

Arquivo Letras.py

Este arquivo contém duas funções `def __init__(self,nome,distancia,estimativa)` sendo esta responsável por receber as letras, a distância e a estimativa de cada letra e a função `def AddLetrasAdjacentes(self,letras)` responsável por adicionar os nós adjacentes no mapa de letras.

```
1 class Letras:
2     def __init__(self,nome,distancia,estimativa):
3         self.nome = nome
4         self.distancia = distancia
5         self.estimativa = estimativa
6         self.visitado = False
7         self.adjacentes = []
8
9     def AddLetrasAdjacentes(self,letras):
10        self.adjacentes.append(letras)
11
```

Arquivo Adjacentes.py

Este arquivo implementa a classe Adjacentes utilizada no método `AddLetrasAdjacentes` para adicionar as letras no mapa.

```
1 class Adjacentes:
2     def __init__(self,letras):
3         self.letras = letras
4
```

Arquivo Mapa.py

Este arquivo é responsável por criar o mapa de letras como parte do desenvolvimento do código importamos os arquivos Letras e Adjacentes e suas classes e métodos para tornar possível a sua implementação.

```

1 from Letras import Letras
2 from Adjacentes import Adjacentes
3
4 class Mapa:
5     A = Letras('A',0,0)
6     B = Letras('B',21,11)
7     K = Letras('K',17,27)
8     J = Letras('J',13,33)
9     L = Letras('L',19,29)
10    M = Letras('M',18,28)
11    C = Letras('C',10,20)
12    G = Letras('G',15,10)
13    H = Letras('H',16,26)
14    N = Letras('N',22,32)
15    O = Letras('O',23,13)
16    D = Letras('D',9,39)
17    E = Letras('E',24,14)
18    F = Letras('F',25,35)
19    R = Letras('R',26,36)
20    S = Letras('S',27,37)
21
22    '''Nós Adjacentes do nó raiz A'''
23    A.AddLetrasAdjacentes(Adjacentes(B))
24    A.AddLetrasAdjacentes(Adjacentes(C))
25    A.AddLetrasAdjacentes(Adjacentes(D))
26
27    '''Nós Adjacentes do nó B'''
28    B.AddLetrasAdjacentes(Adjacentes(K))
29    B.AddLetrasAdjacentes(Adjacentes(J))
30
31    '''Nós Adjacentes do nó C'''
32    C.AddLetrasAdjacentes(Adjacentes(G))
33    C.AddLetrasAdjacentes(Adjacentes(H))
34
35    '''Nós Adjacentes do nó D'''
36    D.AddLetrasAdjacentes(Adjacentes(E))
37    D.AddLetrasAdjacentes(Adjacentes(F))
38
39
40    '''Nós Adjacentes do nó J'''
41    J.AddLetrasAdjacentes(Adjacentes(L))
42    J.AddLetrasAdjacentes(Adjacentes(M))
43
44    '''Nós Adjacentes do nó G'''
45    G.AddLetrasAdjacentes(Adjacentes(N))
46    G.AddLetrasAdjacentes(Adjacentes(O))
47
48
49    '''Nós Adjacentes do nó F'''
50    F.AddLetrasAdjacentes(Adjacentes(R))
51    F.AddLetrasAdjacentes(Adjacentes(S))
52

```

Arquivo VetorOrdenado.py

Este arquivo contém quatro funções responsáveis pela ordenação e busca dos valores do mapa de letras a função `def __init__(self,tamanho)` recebe o tamanho do vetor e o cria, a função `def inserir(self,Letras)` insere os valores no vetor, no caso são as letras, suas distâncias e suas estimativas, estes valores são armazenados no vetor de forma ordenada no sentido crescente, a função `def mostrar(self)` responsável por mostrar os dados do vetor ordenado e a função `def buscar(self)` verifica se a letra digitada pelo usuário existe ou não no mapa de letras, qual a distância (custo) total percorrida para chegar até o nó, qual o valor da estimativa e o valor da função de avaliação.

```
1 class VetorOrdenado:
2     def __init__(self,tamanho):
3         self.numeroElementos = 0
4         self.letras = [None] * tamanho
5
6     def inserir(self,Letras):
7         if self.numeroElementos == 0:
8             self.letras[0] = Letras
9             self.numeroElementos = 1
10            return
11        posicao = 0
12        i = 0
13
14        while i < self.numeroElementos:
15            if Letras.distancia > self.letras[posicao].distancia:
16                posicao += 1
17                i += 1
18
19        for k in range(self.numeroElementos,posicao,-1):
20            self.letras[k] = self.letras[k - 1]
21
22        self.letras[posicao] = Letras
23        self.numeroElementos += 1
24
25    def mostrar(self):
26        print(' Nó  CUSTO ')
27        for i in range(0,self.numeroElementos):
28            print(' {} {} '.format(self.letras[i].nome,self.letras[i].distancia))
29
30
```

```

31 def buscar(self):
32     l = input('Digite a letra para buscar no mapa: ')
33     busca = l.upper()
34     if busca == 'A':
35         custo = self.letras[0].distancia
36         estimativa = self.letras[0].estimativa
37         funcaoAvaliacao = custo + estimativa
38         print('Letra encontrada na posição 0!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
39               '\nValor função avaliação = ',funcaoAvaliacao)
40
41     elif busca == 'B':
42         custo += self.letras[1].distancia
43         custo += self.letras[2].distancia
44         estimativa += self.letras[1].estimativa
45         estimativa += self.letras[2].estimativa
46         funcaoAvaliacao = custo + estimativa
47         print('Letra encontrada na posição 2!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
48               '\nValor função avaliação = ',funcaoAvaliacao)
49
50     elif busca == 'C':
51         custo += self.letras[1].distancia
52         custo += self.letras[2].distancia
53         custo += self.letras[3].distancia
54         custo += self.letras[4].distancia
55         custo += self.letras[5].distancia
56         estimativa += self.letras[1].estimativa
57         estimativa += self.letras[2].estimativa
58         estimativa += self.letras[3].estimativa
59         estimativa += self.letras[4].estimativa
60         estimativa += self.letras[5].estimativa
61         funcaoAvaliacao = custo + estimativa
62         print('Letra encontrada na posição 5!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
63               '\nValor função avaliação = ',funcaoAvaliacao)
64
65     elif busca == 'D':
66         custo += self.letras[1].distancia
67         custo += self.letras[2].distancia
68         custo += self.letras[3].distancia
69         custo += self.letras[4].distancia
70         custo += self.letras[5].distancia
71         custo += self.letras[6].distancia
72         custo += self.letras[7].distancia
73         custo += self.letras[8].distancia
74         custo += self.letras[9].distancia
75         custo += self.letras[10].distancia
76         custo += self.letras[11].distancia
77         custo += self.letras[12].distancia
78         custo += self.letras[13].distancia
79         custo += self.letras[14].distancia
80         custo += self.letras[15].distancia
81         estimativa += self.letras[1].estimativa
82         estimativa += self.letras[2].estimativa
83         estimativa += self.letras[3].estimativa
84         estimativa += self.letras[4].estimativa
85         estimativa += self.letras[5].estimativa
86         estimativa += self.letras[6].estimativa
87         estimativa += self.letras[7].estimativa
88         estimativa += self.letras[8].estimativa
89         estimativa += self.letras[9].estimativa
90         estimativa += self.letras[10].estimativa
91         estimativa += self.letras[11].estimativa
92         estimativa += self.letras[12].estimativa
93         estimativa += self.letras[13].estimativa
94         estimativa += self.letras[14].estimativa
95         estimativa += self.letras[15].estimativa
96         funcaoAvaliacao = custo + estimativa
97         print('Letra encontrada na posição 15!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
98               '\nValor função avaliação = ',funcaoAvaliacao)
99
100
101     elif busca == 'E':
102         custo += self.letras[1].distancia
103         custo += self.letras[2].distancia
104         custo += self.letras[3].distancia
105         custo += self.letras[4].distancia
106         estimativa += self.letras[1].estimativa
107         estimativa += self.letras[2].estimativa
108         estimativa += self.letras[3].estimativa
109         estimativa += self.letras[4].estimativa
110         funcaoAvaliacao = custo + estimativa
111         print('Letra encontrada na posição 4!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
112               '\nValor função avaliação = ',funcaoAvaliacao)
113

```

```

114 elif busca == 'F':
115     custo += self.letras[1].distancia
116     custo += self.letras[2].distancia
117     custo += self.letras[3].distancia
118     custo += self.letras[4].distancia
119     custo += self.letras[5].distancia
120     custo += self.letras[6].distancia
121     custo += self.letras[7].distancia
122     custo += self.letras[8].distancia
123     custo += self.letras[9].distancia
124     custo += self.letras[10].distancia
125     custo += self.letras[11].distancia
126     custo += self.letras[12].distancia
127     estimativa += self.letras[1].estimativa
128     estimativa += self.letras[2].estimativa
129     estimativa += self.letras[3].estimativa
130     estimativa += self.letras[4].estimativa
131     estimativa += self.letras[5].estimativa
132     estimativa += self.letras[6].estimativa
133     estimativa += self.letras[7].estimativa
134     estimativa += self.letras[8].estimativa
135     estimativa += self.letras[9].estimativa
136     estimativa += self.letras[10].estimativa
137     estimativa += self.letras[11].estimativa
138     estimativa += self.letras[12].estimativa
139     funcaoAvaliacao = custo + estimativa
140     print('Letra encontrada na posição 12!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
141           '\nValor função avaliação = ',funcaoAvaliacao)
142
143 elif busca == 'G':
144     custo = self.letras[1].distancia
145     estimativa = self.letras[1].estimativa
146     funcaoAvaliacao = custo + estimativa
147     print('Letra encontrada na posição 1!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
148           '\nValor função avaliação = ',funcaoAvaliacao)
149
150 elif busca == 'H':
151     custo += self.letras[1].distancia
152     custo += self.letras[2].distancia
153     custo += self.letras[3].distancia
154     custo += self.letras[4].distancia
155     custo += self.letras[5].distancia
156     custo += self.letras[6].distancia
157     estimativa += self.letras[1].estimativa
158     estimativa += self.letras[2].estimativa
159     estimativa += self.letras[3].estimativa
160     estimativa += self.letras[4].estimativa
161     estimativa += self.letras[5].estimativa
162     estimativa += self.letras[6].estimativa
163     funcaoAvaliacao = custo + estimativa
164     print('Letra encontrada na posição 6!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
165           '\nValor função avaliação = ',funcaoAvaliacao)
166
167 elif busca == 'J':
168     custo += self.letras[1].distancia
169     custo += self.letras[2].distancia
170     custo += self.letras[3].distancia
171     custo += self.letras[4].distancia
172     custo += self.letras[5].distancia
173     custo += self.letras[6].distancia
174     custo += self.letras[7].distancia
175     custo += self.letras[8].distancia
176     custo += self.letras[9].distancia
177     custo += self.letras[10].distancia
178     custo += self.letras[11].distancia
179     estimativa += self.letras[1].estimativa
180     estimativa += self.letras[2].estimativa
181     estimativa += self.letras[3].estimativa
182     estimativa += self.letras[4].estimativa
183     estimativa += self.letras[5].estimativa
184     estimativa += self.letras[6].estimativa
185     estimativa += self.letras[7].estimativa
186     estimativa += self.letras[8].estimativa
187     estimativa += self.letras[9].estimativa
188     estimativa += self.letras[10].estimativa
189     estimativa += self.letras[11].estimativa
190     funcaoAvaliacao = custo + estimativa
191     print('Letra encontrada na posição 11!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
192           '\nValor função avaliação = ',funcaoAvaliacao)
193
194 elif busca == 'K':
195     custo += self.letras[1].distancia
196     custo += self.letras[2].distancia
197     custo += self.letras[3].distancia
198     custo += self.letras[4].distancia
199     custo += self.letras[5].distancia
200     custo += self.letras[6].distancia
201     custo += self.letras[7].distancia
202     estimativa += self.letras[1].estimativa
203     estimativa += self.letras[2].estimativa
204     estimativa += self.letras[3].estimativa
205     estimativa += self.letras[4].estimativa
206     estimativa += self.letras[5].estimativa
207     estimativa += self.letras[6].estimativa
208     estimativa += self.letras[7].estimativa
209     funcaoAvaliacao = custo + estimativa
210     print('Letra encontrada na posição 7!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
211           '\nValor função avaliação = ',funcaoAvaliacao)

```

```

213 elif busca == 'L':
214     custo += self.letras[1].distancia
215     custo += self.letras[2].distancia
216     custo += self.letras[3].distancia
217     custo += self.letras[4].distancia
218     custo += self.letras[5].distancia
219     custo += self.letras[6].distancia
220     custo += self.letras[7].distancia
221     custo += self.letras[8].distancia
222     custo += self.letras[9].distancia
223     estimativa += self.letras[1].estimativa
224     estimativa += self.letras[2].estimativa
225     estimativa += self.letras[3].estimativa
226     estimativa += self.letras[4].estimativa
227     estimativa += self.letras[5].estimativa
228     estimativa += self.letras[6].estimativa
229     estimativa += self.letras[7].estimativa
230     estimativa += self.letras[8].estimativa
231     estimativa += self.letras[9].estimativa
232     funcaoAvaliacao = custo + estimativa
233     print('Letra encontrada na posição 9!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
234           '\nValor função avaliação = ',funcaoAvaliacao)
235
236 elif busca == 'M':
237     custo += self.letras[1].distancia
238     custo += self.letras[2].distancia
239     custo += self.letras[3].distancia
240     custo += self.letras[4].distancia
241     custo += self.letras[5].distancia
242     custo += self.letras[6].distancia
243     custo += self.letras[7].distancia
244     custo += self.letras[8].distancia
245     estimativa += self.letras[1].estimativa
246     estimativa += self.letras[2].estimativa
247     estimativa += self.letras[3].estimativa
248     estimativa += self.letras[4].estimativa
249     estimativa += self.letras[5].estimativa
250     estimativa += self.letras[6].estimativa
251     estimativa += self.letras[7].estimativa
252     estimativa += self.letras[8].estimativa
253     funcaoAvaliacao = custo + estimativa
254     print('Letra encontrada na posição 8!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
255           '\nValor função avaliação = ',funcaoAvaliacao)
256
257 elif busca == 'N':
258     custo += self.letras[1].distancia
259     custo += self.letras[2].distancia
260     custo += self.letras[3].distancia
261     custo += self.letras[4].distancia
262     custo += self.letras[5].distancia
263     custo += self.letras[6].distancia
264     custo += self.letras[7].distancia
265     custo += self.letras[8].distancia
266     custo += self.letras[9].distancia
267     custo += self.letras[10].distancia
268     estimativa += self.letras[1].estimativa
269     estimativa += self.letras[2].estimativa
270     estimativa += self.letras[3].estimativa
271     estimativa += self.letras[4].estimativa
272     estimativa += self.letras[5].estimativa
273     estimativa += self.letras[6].estimativa
274     estimativa += self.letras[7].estimativa
275     estimativa += self.letras[8].estimativa
276     estimativa += self.letras[9].estimativa
277     estimativa += self.letras[10].estimativa
278     funcaoAvaliacao = custo + estimativa
279     print('Letra encontrada na posição 10!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
280           '\nValor função avaliação = ',funcaoAvaliacao)
281
282 elif busca == 'O':
283     custo += self.letras[1].distancia
284     custo += self.letras[2].distancia
285     custo += self.letras[3].distancia
286     estimativa += self.letras[1].estimativa
287     estimativa += self.letras[2].estimativa
288     estimativa += self.letras[3].estimativa
289     funcaoAvaliacao = custo + estimativa
290     print('Letra encontrada na posição 3!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
291           '\nValor função avaliação = ',funcaoAvaliacao)

```



```

293         elif busca == 'R':
294             custo += self.letras[1].distancia
295             custo += self.letras[2].distancia
296             custo += self.letras[3].distancia
297             custo += self.letras[4].distancia
298             custo += self.letras[5].distancia
299             custo += self.letras[6].distancia
300             custo += self.letras[7].distancia
301             custo += self.letras[8].distancia
302             custo += self.letras[9].distancia
303             custo += self.letras[10].distancia
304             custo += self.letras[11].distancia
305             custo += self.letras[12].distancia
306             custo += self.letras[13].distancia
307             estimativa += self.letras[1].estimativa
308             estimativa += self.letras[2].estimativa
309             estimativa += self.letras[3].estimativa
310             estimativa += self.letras[4].estimativa
311             estimativa += self.letras[5].estimativa
312             estimativa += self.letras[6].estimativa
313             estimativa += self.letras[7].estimativa
314             estimativa += self.letras[8].estimativa
315             estimativa += self.letras[9].estimativa
316             estimativa += self.letras[10].estimativa
317             estimativa += self.letras[11].estimativa
318             estimativa += self.letras[12].estimativa
319             funcaoAvaliacao = custo + estimativa
320             print('Letra encontrada na posição 13!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
321                   '\nValor função avaliação = ',funcaoAvaliacao)
322
323         elif busca == 'S':
324             custo += self.letras[1].distancia
325             custo += self.letras[2].distancia
326             custo += self.letras[3].distancia
327             custo += self.letras[4].distancia
328             custo += self.letras[5].distancia
329             custo += self.letras[6].distancia
330             custo += self.letras[7].distancia
331             custo += self.letras[8].distancia
332             custo += self.letras[9].distancia
333             custo += self.letras[10].distancia
334             custo += self.letras[11].distancia
335             custo += self.letras[12].distancia
336             custo += self.letras[13].distancia
337             custo += self.letras[14].distancia
338             estimativa += self.letras[1].estimativa
339             estimativa += self.letras[2].estimativa
340             estimativa += self.letras[3].estimativa
341             estimativa += self.letras[4].estimativa
342             estimativa += self.letras[5].estimativa
343             estimativa += self.letras[6].estimativa
344             estimativa += self.letras[7].estimativa
345             estimativa += self.letras[8].estimativa
346             estimativa += self.letras[9].estimativa
347             estimativa += self.letras[10].estimativa
348             estimativa += self.letras[11].estimativa
349             estimativa += self.letras[12].estimativa
350             estimativa += self.letras[13].estimativa
351             estimativa += self.letras[14].estimativa
352             funcaoAvaliacao = custo + estimativa
353             print('Letra encontrada na posição 14!\nValor do custo = ',custo,'\nValor da estimativa = ',estimativa,
354                   '\nValor função avaliação = ',funcaoAvaliacao)
355
356         else:
357             print('A letra',busca,'não pertence ao mapa!')
358

```

Arquivo BuscaEstrela.py

Neste arquivo é criada a função def CriarMapaOrdenado() responsável por criar um novo mapa de letras de forma que as estimativas estejam ordenadas no sentido crescente. Também é aplicado o método buscar() que retorna os resultados da busca e mostrar() que apresenta a nova ordenação dos dados do mapa de letras.

```

1 from VetorOrdenado import VetorOrdenado
2 from Mapa import Mapa
3 mapa = Mapa()
4 vetor = VetorOrdenado(16)
5
6 def CriarMapaOrdenado():
7     vetor.inserir(mapa.A)
8     vetor.inserir(mapa.B)
9     vetor.inserir(mapa.C)
10    vetor.inserir(mapa.D)
11    vetor.inserir(mapa.K)
12    vetor.inserir(mapa.J)
13    vetor.inserir(mapa.L)
14    vetor.inserir(mapa.M)
15    vetor.inserir(mapa.G)
16    vetor.inserir(mapa.H)
17    vetor.inserir(mapa.N)
18    vetor.inserir(mapa.O)
19    vetor.inserir(mapa.E)
20    vetor.inserir(mapa.F)
21    vetor.inserir(mapa.R)
22    vetor.inserir(mapa.S)
23
24
25 CriarMapaOrdenado()
26 vetor.buscar()
27 vetor.mostrar()

```

Dados ordenados do MapaOrdenado

| Nó | ESTIMATIVA | CUSTO |
|----|------------|-------|
| A | 0 | 0 |
| G | 10 | 15 |
| B | 11 | 21 |
| O | 13 | 23 |
| E | 14 | 24 |
| C | 20 | 10 |
| H | 26 | 16 |
| K | 27 | 17 |
| M | 28 | 18 |
| L | 29 | 19 |
| N | 32 | 22 |
| J | 33 | 13 |
| F | 35 | 25 |
| R | 36 | 26 |
| S | 37 | 27 |
| D | 39 | 9 |

Realizando buscas no mapa

```

Digite a letra para buscar no mapa: g
Letra encontrada na posição 1!
Valor do custo = 15
Valor da estimativa = 10
Valor função avaliação = 25

```

|| Digite a letra para buscar no mapa: x
|| A letra X não pertence ao mapa!