

Instituto de Computação da UNICAMP

Disciplina MC202: Segundo Semestre de 2014

Laboratório Nº 13

Prof. Tomasz Kowaltowski

O objetivo desta tarefa de laboratório é a implementação do algoritmo de Dijkstra de busca de caminhos mínimos num **grafo orientado** representado por listas de adjacência, conforme foi explicado em aula.

As funções a serem implementadas estão especificadas na interface `dijkstra.h`. O arquivo `dijkstra.c` deve conter a implementação dessas funções bem como a especificação das estruturas necessárias. É fornecido um arquivo incompleto `dijkstra.c` que deve ser completado, testado e submetido.

O programa `principal.c` é fornecido na íntegra e testa as funções de `dijkstra.c`. A entrada de dados do programa compõe-se da descrição do grafo e de uma lista dos vértices a partir dos quais devem ser calculados os caminhos mínimos para todos os outros vértices do grafo. Por exemplo:

```
4
0   2   10
1   2   20
2   1   10
-1  -1  -1
0
1
2
3
-1
```

indica que o grafo tem quatro vértices (primeira linha), com três arestas $0 \rightarrow 2$, $1 \rightarrow 2$ e $2 \rightarrow 1$, com pesos, respectivamente, 10 , 20 , 10 . A linha com três valores -1 é a sentinela que marca o fim das linhas que descrevem as arestas. A seguir vêm a lista dos vértices, um por linha, que devem ser usados como origem para o algoritmo de Dijkstra. Esta lista também termina com uma sentinela.

A saída do programa principal é auto-explicativa e exemplificada nos testes que acompanham a tarefa. No caso do exemplo acima seria:

```
Grafo de 4 vértices:
  0:  2 (10)
  1:  2 (20)
  2:  1 (10)
  3:
Caminhos mínimos a partir do vértice 0:
  1: 20
  2: 10
Caminhos mínimos a partir do vértice 1:
  2: 20
Caminhos mínimos a partir do vértice 2:
  1: 10
Caminhos mínimos a partir do vértice 3:
  nenhum
Processamento terminado
```

Observações:

1. Esta tarefa poderia utilizar uma fila de prioridades semelhante àquela da tarefa 08. Entretanto, seria necessário acrescentar à interface a operação *Sobe* pois os valores contidos na fila podem ter custos decrescentes durante o processo. Sugere-se, portanto, uma implementação mais simples utilizando apenas vetores.
 2. A administração de memória deve utilizar, em lugar das funções `malloc` e `free`, as macros `MALLOC` e `FREE` definidas no pacote `ballocc`.
 3. O número máximo de submissões é 10.
-