

Ficha Prática 07 – Grafos: Árvore de expansão mínima

Instruções

- Faça download do ficheiro **cal_fp07.zip** da página da disciplina e descomprima-o (contém os ficheiros **Test.cpp** e **Graph.h**, com código no seguimento das aulas práticas anteriores)
- Abra o eclipse e crie um novo projeto C++ do tipo Cute Project (File/New/C++ Project/Cute Project) com o nome **CalFp07** e usando o MinGW GCC.
- Inclua a biblioteca Boost.
- Importe para a pasta *src* do projeto, os ficheiros extraídos: (Import/General/File System)
 - Aparecendo a mensagem a perguntar se quer fazer *Overwrite* ao ficheiro Test.cpp diga que sim.
 - Compile o projeto.
 - Execute o projeto como CUTE Test (Run As/CUTE Test). Se surgir a pergunta de qual compilador usar, escolha MinGW gdb.
- **Deverá realizar esta ficha respeitando a ordem das alíneas.** Poderá executar o projeto como CUTE Test quando quiser saber se a implementação que fez é suficiente para passar no teste correspondente.

Enunciado

Considere a classe **Graph** definida no ficheiro *Graph.h* e já utilizada nas aulas anteriores. Deverá atualizar as classes do ficheiro *Graph.h* adequadamente, a fim de realizar as alíneas que se seguem. Identifique a partir do ficheiro Test.cpp funções auxiliares que sejam necessárias e não sejam pedidas explicitamente nos exercícios.

a) Implemente na classe **Graph** o membro-função público:

```
vector<Vertex<T>*> calculatePrim()
```

Esta função implementa o algoritmo de Prim para encontrar a árvore de expansão mínima a partir do primeiro vértice v do grafo a todos os outros vértices.

b) Implemente na classe **Graph** o membro-função público:

```
vector<Vertex<T>*> calculateKruskal()
```

Esta função implementa o algoritmo de Kruskal para encontrar a árvore de expansão mínima no grafo.