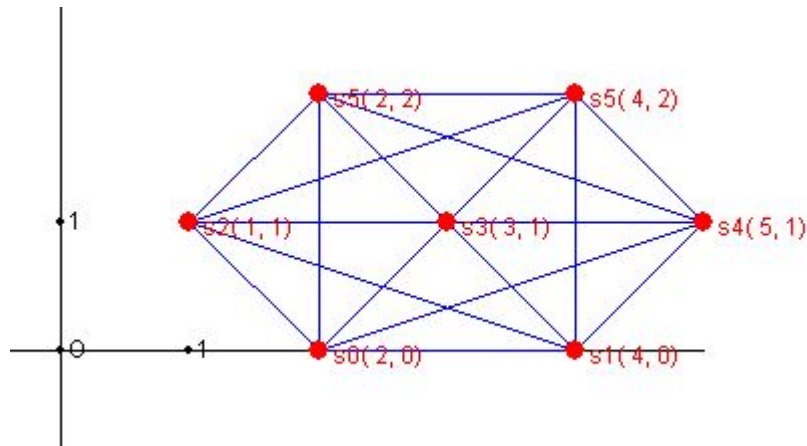


Soutenance projet TSP et recuit simulé

Chercher une solution au TSP par l'algorithme du recuit simulé vu en TP pour le graphe suivant :



Le code source du programme principal est déjà fourni partiellement (cf. clé USB fournie)

- 1) afficher en mode console la solution
- 2) afficher la longueur du cycle trouvé
- 3) dessiner le graphe et la solution trouvée
- 4) extensions ?

```

/**
Construction de graphes qui peuvent servir d'exemples d'application de l'algorithme du recuit
simulé.
Cet algorithme est utilisé pour trouver une solution au problème du voyageur de commerce
*/

#include <fstream>
#include <iostream>
#include <string>

#include "Graphe.h"
#include "DessinGrapheRecuitSimule.h"
#include "OutilsCarteRecuitSimule.h"

using namespace std;

#define S1 7 // nombre de sommets du graphe g1
#define A1 (S1*(S1-1)/2) // nombre d'arêtes du graphe g1 car g1 est complet

//int main9()
int main()
{
    char ch;
    { //----- 1er exemple de graphe -----

        cout << "construction d'un 1er exemple de graphe complet à 7 sommets" << endl;

        Graphe<InfoAreteCarte,InfoSommetCarte> g1; // création du graphe g1 vide

        //----- on crée les sommets dans g1 -----

        Sommet<InfoSommetCarte> * s[S1];
        s[0] = g1.creeSommet(InfoSommetCarte("s0",Vecteur2D(2,0)));
        s[1] = g1.creeSommet(InfoSommetCarte("s1",Vecteur2D(4,0)));
        s[2] = g1.creeSommet(InfoSommetCarte("s2",Vecteur2D(1,1)));
        s[3] = g1.creeSommet(InfoSommetCarte("s3",Vecteur2D(3,1)));
        s[4] = g1.creeSommet(InfoSommetCarte("s4",Vecteur2D(5,1)));
        s[5] = g1.creeSommet(InfoSommetCarte("s5",Vecteur2D(2,2)));
        s[6] = g1.creeSommet(InfoSommetCarte("s5",Vecteur2D(4,2))); // cette répétition de
        6 instructions pourrait être avantageusement remplacée par une boucle

        // elle a été uniquement été laissée pour améliorer la lisibilité
    }
}

```

```

//----- on crée les arêtes dans g1 -----

Arete<InfoAreteCarte,InfoSommetCarte> * a[A1];

int i,j; // indices des sommets
int k;    // indice de l'arête courante

for (i = 0, k = 0; i < S1; ++i)
    for (j = i+1; j < S1; ++j)
        {
            double d = OutilsCarteRecuitSimule::distance(s[i],s[j]); // calcul de la distance
            du sommet s[i] à s[j]
            a[k++] = g1.creeArete( s[i], s[j], InfoAreteCarte(d));

            // on peut remplacer les 2 lignes précédentes par l'unique ligne suivante :
            // a[k++] = OutilsCarteRecuitSimule.creeArete(s[i],s[j],g1);
        }
//----- ca y est, g1 est créé et complet -----

// ----- on affiche sur la console toutes les informations contenues dans g1

cout << "g1 = "<< endl << g1 << endl;

cout <<"tapez un caractère, puis ENTER\n"; cin >> ch;

//----- on crée le fichier texte pour dessiner g1 -----

string nomFichierDessin = "grapheHeptagonalComplet.txt";
ofstream f(nomFichierDessin); // ouverture
de f en écriture, en mode texte (cf. doc cplusplus.com)
Vecteur2D coinBG(-1,-1), coinHD(5,5); // limites de la
fenêtre à visualiser. calculées à partir des coordonnées des sommets
string couleurRepere = "blue";
int rayonSommet = 5; // unité :
pixel
string couleurSommets = "red";
string couleurAretes = "blue";

DessinGrapheRecuitSimule::ecritGraphe(f, g1, coinBG, coinHD, couleurRepere,
rayonSommet, couleurSommets, couleurAretes);

cout << "le fichier texte de dessin " << nomFichierDessin << " a été créé"<< endl;

} //----- fin 1er exemple de graphe -----

cin >> ch;
return 0;
}

```