Algorithmique des Graphes
L3 informatique - Université de Toulon

5 janvier 2021

Vous êtes invités à remettre une copie claire, concise, sans rature ni surcharge. Il est inutile de recopier l'énoncé... La note finale tiendra compte de la présentation générale de la copie.

Questions de cours

Q1. Quel est le nombre d'arêtes maximal d'un graphe d'ordre n ? Dans ce cas, comment nomme-t-on le graphe ?

C'est le nombre de parties à 2 éléments dans un ensemble à n élément, soit $n(n-1)/2$. Il s'agit du graphe complet K_n.

Q2. Pour une partie S de l'ensemble des entiers naturels, quelle est la définition de $\text{mex}(S)$?

$\text{mex}(S)$ est le plus petit entier n'appartenant pas à S.

Q3. Quelle structure est utilisée pour l'implantation de la gestion des ensembles disjoints avec l'heurisique de l'union par rang ? Préciser le temps de calcul de l'opération REPRESENTANT, en fonction du nombre n d'opérations SINGLETON.

```
typdef struct _ed_ {
 int rang;
 struct _ed_ * rep;
}
```

Le temps de calcul est $O(\log n)$.

Q4. Donner une définition de la notion de *tri topologique* du point de vue de la théorie algorithmique des graphes, en précisant bien le contexte et le temps de calcul. Citer un exemple d'application.

Le tri topologique d'un DAG consiste à ordonner les sommets de sorte que si u précède v alors il n'existe pas d'arc de v vers u.

Complexité de Kruskal

Q5. On considère l'algorithme de Kruskal procédant sur un graphe complet d'ordre n. Donner le temps de calcul en fonction n suivant l'algorithme de tri utilisé :

1. tri linéaire
2. tri par sélection
3. tri rapide

Le coût sur la structure d'ensembles disjoints est au plus $O(n^2 \log n)$, il faut ajouter le coût du tri, le bilan suivant les cas est :

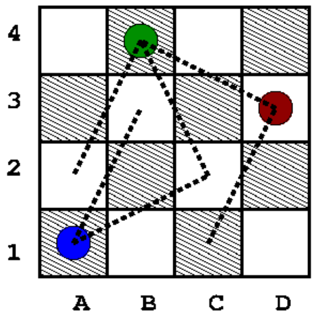
1. (linéaire) $\theta(n^2 \log n)$;
2. (sélection) $\theta(n^4)$;
3. (rapide) de l'ordre de $n^2 \log n$.

création algorithme

Q6. Proposer une méthode efficace pour déterminer le plus petit entier ne figurant pas parmi les valeurs d'un tableau de n entiers. Préciser le temps de calcul.

Le plus petit de ces entiers est obligatoirement inférieur ou égal à n. On peut prendre un tableau de taille $n+1$ initialisé à 0, puis on marque les valeurs du tableau qui sont strictement inférieures n. On détermine la plus petite position non marquée, c'est la valeur recherchée. La complexité est linéaire.

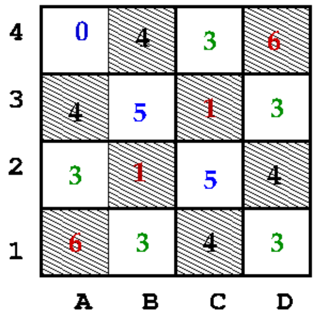
Déplacement du Cavalier



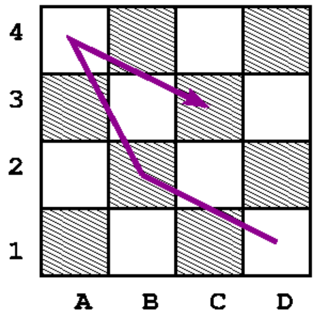
Q7. On considère le graphe des déplacements d'un cavalier sur les 16 cases d'un échiquier 4×4. Par exemple, la case $A1$ est adjacentes aux cases $C2$ et $B3$.

1. Le graphe est-il connexe ?
2. Possède-t-il un cycle de longueur impair?
3. Le graphe est-il eulérien ?
4. Possède-t-il un cycle hamiltonien ?
5. Le graphe est-il planaire ?

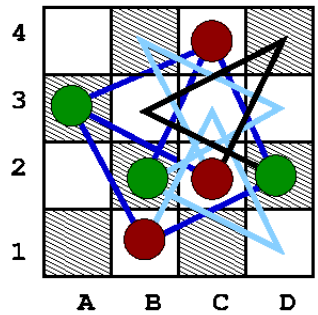
1. Le graphe est-il connexe ? Oui, en partant de la case a4, on peut atteindre tous les autres cases par un chemin de longueur au plus 6.



2. Possède-t-il un cycle de longueur impair? Non, car bicolore.
3. Le graphe est-il eulérien ? Pas de sommets impairs et connexe donc eulérien.
4. Possède-t-il un cycle hamiltonien ? Non, la démontration a été faite en cours. Mais ici, on peut raisonner sur la figure ci-dessous. Aux symétries près, un cycle Hamiltonien passe par le chemin indiqué. En quittant la case $C3$, la case $D1$ devient inaccessible.



5. Le graphe est-il planaire ? Non, car c'est une expansion du graphe $K_{3,3}$.



Graphe de quadruplets

On note $[0, t[$ l'intervalle des entiers naturels strictement inférieurs à t. On considère un quadruplet d'entiers $p := (a, b, c, d)$, paramètre auquel on associe le produit des intervalles :

$$\mathfrak{Q} = [0, a[\times [0, b[\times [0, c[\times [0, d[$$

Notons que $p \notin \mathfrak{Q}$, et que le quadruplet (i, j, k, l) est dans $\mathfrak{Q}$ si et seulement si

$$0 \leq i < a, \quad 0 \leq j < b; \quad 0 \leq k < c, \quad 0 \leq l < d.$$

On considère le graphe Γ dont $\mathfrak{Q}$ est l'ensemble des sommets; en traçant un arc de r vers r' si et seulement si le quadruplet r' se déduit de r en diminuant strictement une seule des composantes de r. On utilisera la notation $r \rightsquigarrow r'$. Par exemple, $(1, 2, 3, 4) \rightsquigarrow (1, 2, 1, 4)$ mais $(1, 2, 3, 4) \not\rightsquigarrow (1, 2, 1, 1)$ et $(1, 2, 3, 4) \not\rightsquigarrow (1, 2, 3, 4)$.

Q8. Préciser la nature du graphe Γ ?

Un graphe orienté acyclique.

Q9. Exprimer l'ordre de Γ en fonction de a, b, c et d.

L'ordre du graphe est $abcd$.

Q10. Quel est le nombre d'arcs Γ ?

$$\sum_{i=0}^{a} + \sum_{j=0}^{b} j + \sum_{k=0}^{c} k + \sum_{l=0}^{d} l$$
$$= \frac{1}{2}(a^2 + b^2 + c^2 + d^2 + a + b + c + d)$$

Q11. Quel est le degré du sommet (i, j, k, l) ?

Le degré est $i + j + k + l$.

PRÉLIMINAIRES ARITHMÉTIQUES

Q12. Effectuer la division euclidienne de l'entier naturel a par l'entier $b > 0$ c'est déterminer l'unique couple d'entiers (q, r) vérifiant :

$$a = bq + r, \quad 0 \leq r < b.$$

1. Comment s'appelle les quantités q et r ?
2. En langage C, quels sont les opérateurs permettant de calculer q et r ?

Le quotient (a/b) et le reste (a%b).

Q13. Montrer que l'application :

$$(i, j, k, l) \mapsto \mathbf{N}(i, j, k, l) := labc + kab + ja + i$$

est une bijection de $\mathfrak{Q}$ sur l'intervalle $[0, abcd[$.

Si $\mathbf{N}(i, j, k, l) = \mathbf{N}(i', j', k', l')$ alors

$$labc + kab + ja + i = l'abc + k'ab + j'a + i'$$

Comme i et i' sont inférieurs à a, la réduction modulo a donne $i = i'$. On en déduit

$$lbc + kb + j = l'bc + k'b + j'$$

et pour des raisons similaires $j = j'$, puis $k = k'$ et $l = l'$. L'application est donc injective. La valeur maximale est :

$$(d-1)abc+(c-1)ab+(b-1)a+(a-1) = abcd-1.$$

et donc les $abcd$ premiers entiers sont atteints par $\mathbf{N}$ (surjection).

Q14. Dans le cas, $a = 5$, $b = 7$, $c = 9$ et $d = 11$. Quelle est l'image de $(1, 2, 3, 4)$ par $\mathbf{N}$? Quel est l'antécédant de 2021?

$$1 + 2 * 5 + 3 * 7 * 5 + 4 * 5 * 7 * 9 = 1376$$

et l'antécédant (i, j, k, l) de 2021 s'obtient par réducions successives : $i = 1$, $j = 5$, $k = 3$ et $l = 6$. On a bien :

$$1 + 5 * a + 3 * a * b + 6 * a * b * c = 2021$$

MISE EN OEUVRE

On définit les types :

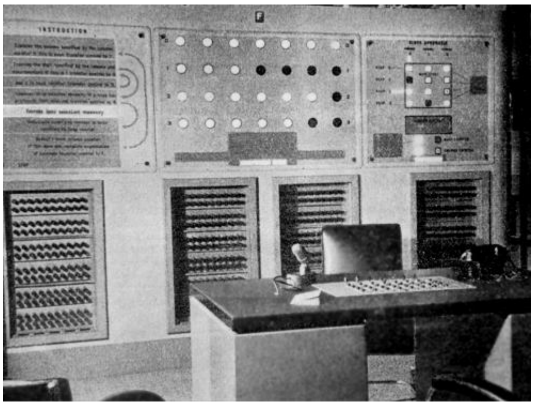
```c
    typedef int quad[4]; // quadruplet
    typedef struct {
        int nbs;          // ordre du graphe
        char ** mat;  // mat. adjacence
    } graphe;
```

Q15. Coder `int itoq( int t, quad r, const quad p)` qui détermine la valeur de r de sorte que $\mathbf{N}(r) = t$ pour le paramètre p. La valeur booléenne de retour renseigne sur la validité du résultat.

```c
    int itoq( int t, quad r, quad p )
    {
        r[0] = t % p[0]; t = t / p[0];
        r[1] = t % p[1]; t = t / p[1];
        r[2] = t % p[2]; t = t / p[2];
        r[3] = t;
        return t < p[3];
    }
```

Q16. Coder `graphe nimrod( quad p )` une fonction qui retourne le graphe Γ correspondant au paramètre p.

```c
graphe nimrod( quad p )
{ quad r;
  int i, j,k, l;
  sommet s;
  graphe g = initGraphe( p[0]*p[1]*p[2]*p[3] );
  for( s = 0; s < g.nbs ) {
    itoq( s, r, p );
    for( i = 0 ; i < r[0]; i++ )
        g.mat[t][ qtoi( i, r[1], r[2], r[3], p ) ] = 1;
    for( j = 0 ; j < r[1]; j++ )
        g.mat[t][ qtoi( r[0], j , r[2], r[3], p ) ] = 1;
    for( k = 0 ; k < r[2]; k++ )
        g.mat[t][ qtoi( r[0], r[1], k, r[3], p ) ] = 1;
    for( l = 0 ; l < r[3]; l++ )
        g.mat[t][ qtoi( r[0], r[1] , r[2], l, p ) ] = 1;
    }
  return g;
}
```



The NIMROD (1951) was designed exclusively to play the game of 'NIM'. This is a simple game, where you start with a number of piles of tokens — traditionally matches. Each player in turn takes one or more tokens from any one pile, and the game continues until the last token is taken from the last remaining pile. .