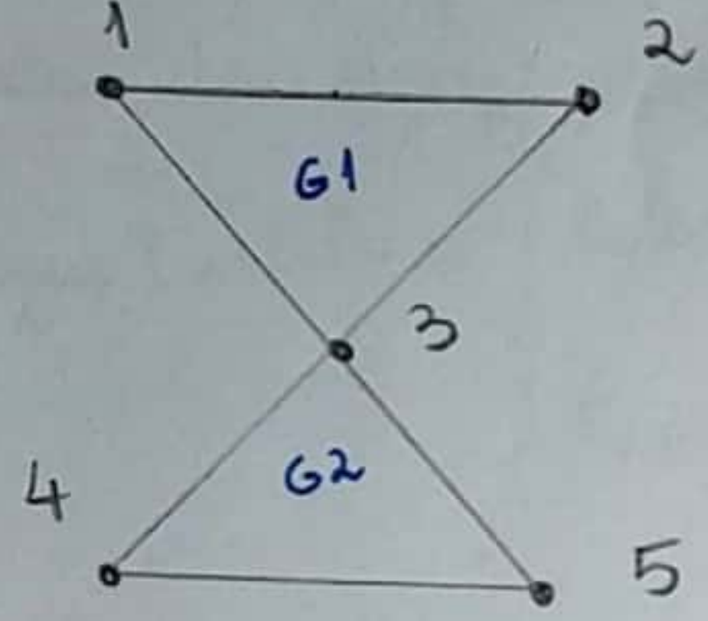
README

Pentru găsirea unei transformări polinomiale T, a problemei kClique am considerat că într-un graf complet, toate nodurile sunt unite prin muchii.

Adică :



Corespunzător fiecărei muchii avem relația de implicație logică - $x_1 \to x_2$ a cărui tabelă de adevăr este :

x_1	x_2	$x_1 \to x_2$
0	0	1
0	1	1
1	0	0
1	1	1

$$x_1 \to x_2 \iff \sim x_1 \vee x_2$$

Așadar funcția polinomială corespunzătoare unei muchii din graf este $\sim x_1 \vee x_2$ și pentru graful G_1, funcția polinomială este :

$$G = (\sim x_1 \vee x_2) \wedge (\sim x_2 \vee x_3) \wedge (\sim x_1 \vee x_3)$$

G verifică valoarea true pentru $x_1 = F$, $x_2 = T$, $x_3 = T$

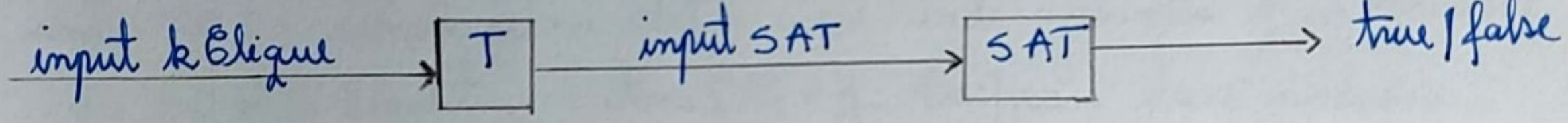
Pentru generarea clauzelor am considerat condițiile de existență a muchiilor (relația de implicație) și am construit polinomul doar în cazurile în care kClique întoarce "True".

Scanned with CamScanner

Eliqua este în NP așa că poate fi redusă la SAT folosind spațiu logaritmic. kEliqua este în P pentru fiecare k fixat deci ar trebui să existe o reducere eficientă

Considerăm funcția bool is-clique(). Funcția verifică dacă un set de noduri din array-ul store formează un clique (subgraf complet) sau nu. Funcția primește ca parametru b ce reprezintă numărul de noduri din array-ul store.

$$\text{bool is\_clique(int b)} = \begin{cases} \text{false, dacă există în matricea graph o valoare 0} \\ \text{true, altfel} \end{cases}$$



Transformarea:

Citim din fișierul de intrare linie cu linie
- citim k - dimensiunea clicii căutate
- citim N - numărul de noduri din graf
- citim M - numărul de muchii din graf

Creăm o matrice edges de dimensiune M x 2 - M linii și 2 coloane
Citim în ea nodurile din fișierul de intrare, fiecare linie a matricii constituind o muchie între nodurile edge[i][0] și edge[i][1]
După ce s-a terminat citirea putem închide fișierul de intrare.
Dacă dimensiunea clicii căutate nu este 0 (k != 0) construim:
1) O matrice pătratică graph de dimensiune N x N cu valori binare astfel încât la pozițiile [edges[i][0]][edges[i][1]] și [edges[i][1]][edges[i][0]] se află valoarea 1, 0 în rest.
2) Un vector d ce are ca indici nodurile grafului, aflați în matricea edges (1, 2, ... N). În acest vector contorizăm numărul

de muchii al fiecărui nod.
Construcțiile se opresc când i-ul ajunge la valoarea size-1.
Variabila size reține valoarea dată de expresia sizeof(edges)/sizeof(edges[0]
ce reprezintă numărul de elemente al matricei edges.

Definim o variabilă booleană gata inițializată cu 0.
Apelăm funcția findCliques() prin apelul findCliques(0, 1, k)
0, 1 reprezintă valorile inițiale ale indecșilor de căutare
Funcția void findCliques(int i, int l, int s) caută primul clique de dimensiune s și dacă îl găsește variabila gata = 1.
Se observă că dimensiunea clicii s determină ca numărul minim de muchii ale unui nod să fie s-1.
Astfel parcurgând vectorul d și ținând cont de observația anterioară, construim un vector stare unde adăugăm doar nodurile care satisfac condiția: stare[l] = j. Verificăm dacă nodurile din stare formează o clică și dacă nu s-a atins dimensiunea clicii completăm în continuare vectorul prin apelarea recursivă a funcției findClique().

Când se ajunge la dimensiunea clicii se construiește funcția polinomială satisfiabilă corespunzător clicii găsite cu funcția print_sat1().
Dacă gata rămâne 0 înseamnă că nu s-a găsit nicio clică și se construiește o funcție polinomială nesatisfiabilă.

Complexitatea algoritmului este $O(n^2 \cdot k)$ ce denotă un timp polinomial de rulare.

Task 3

În cadrul testelor întâlnim 3 tipuri de teste.

Prima categorie conține un k destul de mare în comparație cu N Numărul de muchii este relativ mic raportat la ideea de clique care reprezintă un subgraf complet al grafului inițial. Numărul total de muchii este $\frac{(n-1)n}{2}$.

A doua categorie conține un k foarte mic în comparație cu numărul de noduri și muchii. O dată cu determinarea unui clique în acest caz se poate deduce (sau aștepta) un timp de rulare mai mare.

A treia categorie conține un k destul de mare în comparație cu numărul de noduri și muchii. Deoarece k-ul este mai apropiat ca valoare de N, determinarea unui clique de dimensiune apropiată grafului este așteptată. Acest aspect implică și faptul că numărul de muchii al grafului respectiv al clicii este apropiat.

Timpi de rulare

categ.	BKT	RAT	BKT/RAT
1	0.011s	0.435s	39.545
2	0.021s	0.778s	37.047
3	0.030s	1.200s	40.000

În categoria 2, raportul este cel mai mare datorită k-ului mic. Pentru un k mare în raport cu N, eficiența SAT crește și BKT scade

Scanned with CamScanner