

#### Tema 4 Analiza Algoritmilor

Am realizat transformarea din exemplul dat in enuntul temei.

1) Se folosesc  $N * k$  variabile ( $N = \text{nr de noduri}$ ,  $k = \text{nr de noduri de ales pentru problema k Vertex Cover}$ ). Se va adauga o clauza pentru fiecare muchie, realizand sau intre toti literalii celor 2 noduri. Aceasta operatie este realizata de functia `writeEdges()`.

2) Se verifica daca exista un Vertex Cover de fix  $k$  elemente. Se formeaza multimi de  $N$  literali si se doreste ca exact unul sa fie adevarat cu clauze de tip `at_least_one` si `at_most_one`. Acestea sunt de forma  $(X_1 \vee X_2 \vee \dots \vee X_n) \wedge (\neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee \neg X_3) \wedge \dots$ , clauzele formate din 2 literali sunt combinari de  $N$  luate cate 2. Se verifica toate cele  $k$  multimi de  $N$  literali. Aceasta operatie este realizata de functia `oneTrue()`.

3) Pentru fiecare literal al fiecarui nod, se verifica ca doar unul sa fie adevarat, pentru a garanta ca un nod nu a fost ales de 2 ori. Clauzele sunt de forma  $(\neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee \neg X_3) \wedge \dots$ , reprezentand  $N$  combinari de  $k$  luate cate 2. Aceasta operatie este realizata de functia `oneVariableTrueForNode()`.

Complexitatea

$k$  are cel mult valoarea  $N$

$M$  este numarul de muchii, care poate fi cel mult  $N-1$

1) Complexitatea lui `writeEdges()` este  $O(k * N * M)$ , ceea ce inseamna ca rezultatul final este  $O(n^3)$ .

2) Complexitatea lui `oneTrue()` este  $O(k * N! / ((N-2)! * 2!))$ , de unde rezulta dupa efectuarea simplificariilor  $O((k * (N-1) * N) / 2)$ , deci complexitatea  $O(n^3)$ .

3) Complexitatea lui `oneVariableTrueForNode()` este  $O(N * k! / ((k-2)! * 2!))$ , de unde rezulta dupa efectuarea simplificariilor  $O(((k-1) * k * N) / 2)$ , deci complexitatea  $O(n^3)$ .

Complexitatea transformarii:  $O(n^3) \Rightarrow$  transformare polinomiala

Daia Gabriel Petru 321CB

Username hackerrank: `gabriel_petru_d1`