

Esercizio. Consideriamo un insieme di n variabili x_1, \dots, x_n . Sono dati un insieme di vincoli di uguaglianza della forma $x_i = x_j$, e un altro insieme di vincoli di disuguaglianza della forma $x_i \neq x_j$. Il problema consiste nel capire se tutti i vincoli possono essere soddisfatti. Ad esempio, considerando quattro variabili x_1, x_2, x_3, x_4 soggette ai vincoli $x_1 = x_2; x_2 = x_3; x_3 = x_4; x_1 \neq x_4$ risulta che in questo caso i vincoli non sono soddisfacenti. Progettare un algoritmo che, dati in input il numero n di variabili e le liste dei vincoli di uguaglianza e disuguaglianza, restituisce *true* se i vincoli sono soddisfacenti, *false* altrimenti.

Soluzione. Per risolvere il problema si può utilizzare una struttura Union-Find per rappresentare gli insiemi degli indici delle variabili che devono essere fra loro uguali. Inizialmente si crea un singoletto per ogni indice, eseguendo *makeSet*(i) per $1 \leq i \leq n$. Successivamente, per ogni vincolo di uguaglianza si fa la *union* degli insiemi a cui appartengono gli indici delle variabili corrispondenti. Poi si verifica che i vincoli di disuguaglianza siano tra variabili in insiemi diversi. L'algoritmo scritto in pseudocodice è riportato in tabella Algorithm 1.

Algorithm 1: VERIFICA(*intero* n , *Lista*[*Coppie*] *uguali*, *Lista*[*Coppie*] *diverse*) \rightarrow *Booleano*

Input: Il numero n di variabili, e due liste di coppie di indici, rispettivamente quelli delle variabili uguali, e quelli delle variabili diverse

Output: Un booleano: *true* se i vincoli sono soddisfacenti, *false* altrimenti

UnionFind *UF*

// Creazione di un singoletto per ogni indice di variabile

for $i \leftarrow 1$ **to** n **do**

 | *UF.makeSet*(i)

// Si uniscono insiemi di indici di variabili uguali

for each $(i, j) \in \text{uguali}$ **do**

 | *UF.union*(*UF.find*(i), *UF.find*(j))

// Se due indici di variabili diverse sono nello stesso insieme, i vincoli sono insoddisfacenti

for each $(i, j) \in \text{diversi}$ **do**

 | **if** *UF.find*(i) = *UF.find*(j) **then**

 | **return** *false*

// Se tutti gli indici di variabili diverse appartengono a insiemi distinti, i vincoli sono soddisfacenti

return *true*

Tale algoritmo esegue n operazioni *makeSet*, una quantità $O(u)$ di *union* e di *find*, con u numero di vincoli di uguaglianza, ed una quantità $O(d)$ di *find*, con d numero di vincoli di disuguaglianza. La dimensione degli insiemi su cui si eseguono le *union* e le *find* è $O(n)$. Se adottassimo un'implementazione della struttura Union-Find tramite tecnica QuickUnion con euristica sul rango, il tempo di esecuzione dell'algoritmo sarebbe $T(n, u, d) = \Theta(n) + O(u \log n) + O(d \log n)$. Nel caso in cui $u = O(n)$ e $d = O(n)$, avremmo $T(n) = O(n \log n)$.