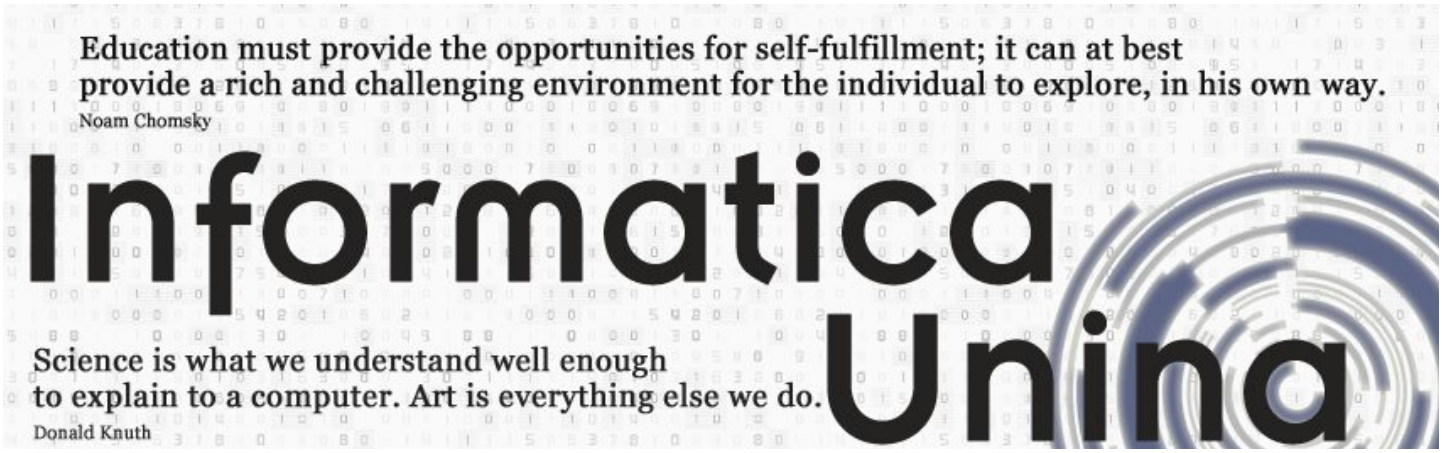


INFORMATICA UNINA

Gruppo Privato · 2307 membri

Invita

Vincenzo Tramo ha fatto una domanda .
13 marzo alle ore 13:42 ·

[ASD] Qualcuno mi può dare una mano a risolvere questo esercizio?

Tema d'esame di Algoritmi e Strutture Dati I
13/01/2021

1. Sia dato un grafo orientato $G = \langle V, E \rangle$, rappresentato con liste di adiacenza, e un vertice s e due insiemi di vertici $B \subseteq V$ e $C \subseteq V$, rappresentati come array. Si scriva un algoritmo che, dati in ingresso G, s, B e C, collezioni in **tempo lineare sulla dimensione di** G in una lista L tutti i vertici v che soddisfano entrambe le seguenti condizioni:
   - v appartiene a B e può raggiungere s tramite un percorso;
   - esiste anche un percorso da s a v che non passa per alcun vertice di C.

2 5 risposte

Mi piace Commenta

Tutti i commenti

Giuseppe Amato
Io farei così (se trovi qualche incongruenza fammi sapere): coloro di "rosso" tutti i vertici di C in modo che non saranno mai attraversati (poichè come sappiamo la DFS o BFS attraverserà solo i nodi bianchi), faccio il grafo trasposto di G (che chiameremo Gt), faccio partire una visita (sia essa BFS o DFS è indifferente), da s (in Gt) e tutto ciò che sarà raggiungibile diventerà nero. Alla fine vado a scorrere l'array B e vedo quali sono i nodi neri. Ora siccome abbiamo fatto il trasposto sapremo che i vertici di B che sono neri avramno dei percorsi, nel grafo originale, che andranno a finire in s e saremo sicuri che se questi sono stati colorati di nero allora ci sono arrivato senza passare per vertici di C poichè colorandoli diversamente da bianco non possono essere attraversati. Non so se sono stato chiaro, in qualunque caso fammi sapere

Mi piace · **Rispondi** · 1 sett. · Modificato 1

Vincenzo Tramo Autore
Ciao può essere pure che mi sbagli però in questo modo conosci solo i vertici che possono essere raggiunti dalla sorgente s data in input senza passare per i vertici di C
La traccia richiede anche che i vertici abbiano un percorso da v alla sorgente s ... **Altro...**

Mi piace · **Rispondi** · 1 sett.

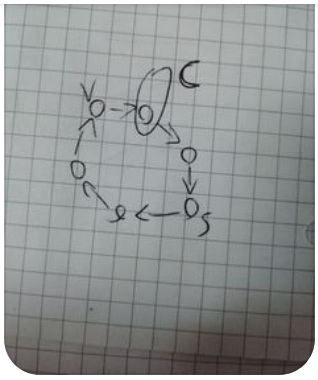
Giuseppe Amato
Ciao, no in realtà dovremmo avere tutta l'informazione che ci serve. Ricorda che la visita a partire da s la facciamo nel grafo trasposto, tale visita colorerà di nero tutti quei vertici v che grafo originale (non quello trasposto) raggiungono s (quest... **Altro...**

Mi piace · **Rispondi** · 1 sett. · Modificato

Antonio Nardella
però colorando già di rosso i vertici di C rischi che magari s non raggiunge v nel grafo trasposto ma esiste comunque un percorso da s a v che nn passa da C

Mi piace · **Rispondi** · 1 sett.

Antonio Nardella
come in questa foto (qui v raggiunge s e s raggiunge v senza passare per C)



Mi piace · **Rispondi** · 1 sett. · Modificato

Giuseppe Amato
Sisi vero, rileggendo la traccia mi sono accorto di aver confuso la seconda condizione

Mi piace · **Rispondi** · 1 sett.

Scrivi una risposta...

Pasquale Barbaro

```
ALGO(G, s, B, C)
INIT(G)
List L;

for v in C do
color[v] = black
endfor

DFS_VISIT(G,s)

for v in B do
if (color[v] = black) then
L.insertAtFront(v)
endif
endfor

INIT(G)
TRASPOSTO(G)

DFS_VISIT(G,s)

for v in B do
if (color[v] = white) then
L.remove(v)
endif
endfor

return L
end ALGO
```

Mi piace · **Rispondi** · 1 sett.

Vincenzo Tramo Autore
Giuseppe Amato Ciò che hai suggerito da solo una parte dell'informazione che ci serve:

Quello di cui abbiamo bisogno richiede di più, in particolare il problema chiede i vertici che fanno parte dell'intersecazione tra questi due insiemi:

- L'insieme dei vertici di B raggiungibili da s senza passare per alcun vertice di C.
(BFS, l'informazione viene salvata in un array COLORE)

- L'insieme dei vertici di V che raggiungono s tramite un percorso.
Nel grafo trasposto, questo equivale a trovare l'insieme dei vertici di V raggiungibili da s.
(Allora utilizziamo il grafo trasposto per usare un'altra BSF2 e tale informazione viene salvata in un altro array COLORE2)

A questo punto facciamo l'intersecazione tra gli array. Gli elementi di B che soddisfano:
(COLORE[b]=COLORE2[b]='nero')
sono i vertici che cerchiamo e allora li andiamo ad inserire nella lista.

Questo è l'algoritmo che ho implementato
Il tutto viene fatto in tempo lineare sulla dimensione del grafo fatemi sapere se vi trovate

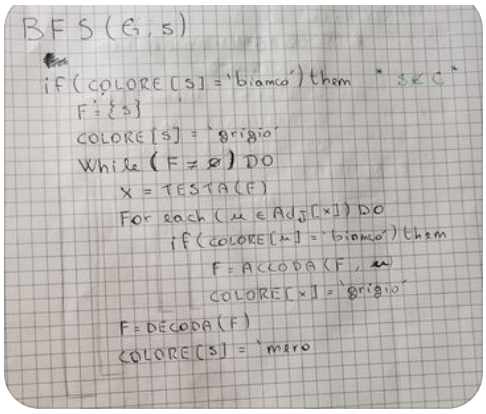
Mi piace · **Rispondi** · 1 sett. · Modificato

Giuseppe Amato
Vincenzo Tramo Si, hai ragione, andando a rileggerr la traccia mi sono accorto di aver letto male la seconda condizione, anche lì avevo letto che v doveva raggiuntere s e non il contrario. Si, dovrei trovarmi con te 😉

1

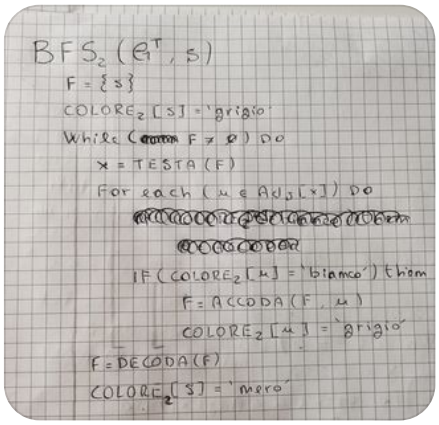
Mi piace · **Rispondi** · 1 sett.

Vincenzo Tramo Autore



Mi piace · **Rispondi** · 1 sett.

Vincenzo Tramo Autore



Mi piace · **Rispondi** · 1 sett.

Scrivi una risposta...