Algoritmi & Laboratorio, Grafi
Docente: András Horváth
Esame del 5 aprile 2006

1. (6 punti) La visita in profondità di un grafo orientato risulta nel seguente sequenza di inizio/fine visita: 1/14, 2/5, 3/4, 6/13, 7/8, 9/10, 11/12, 15/24, 16/23, 17/18, 19/20, 21/22. Il grafo contiene cicli di cui uno è di lunghezza 7.

- Si disegni la foresta di scoperta della visita in profondità.
- Si indichi il ciclo di lunghezza 7.
- Per ognuno dei seguenti archi si specifichi, spiegando il perchè, se possono essere presenti nel grafo visitato:
  - 6/13 $\rightarrow$ 16/23
  - 7/8 $\rightarrow$ 3/4
  - 17/18 $\rightarrow$ 15/24
  - 16/23 $\rightarrow$ 6/13

2. (3 punti) Si dimostri il seguente teorema: se in un grafo orientato ogni nodo contiene almeno un arco entrante allora il grafo è ciclico.

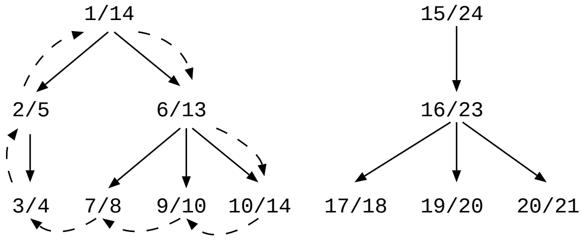
3. (7 punti) Si riporti l'algoritmo di Kruskal modificato in modo tale che esso trovi l'albero di copertura massimo invece di quello minimo. Si applichi l'algoritmo al grafo riportato sotto. Ogni volta che un arco viene aggiunto alla soluzione, si indichino i due insiemi di nodi che vengono uniti.

Il grafo è dato tramite le lista di adiacenti (il numero tra parentesi è il peso del arco):

- A: B(3), D(7), G(9)
- B: A(3), C(2), D(4), E(11), F(8)
- C: B(2), E(10), I(1)
- D: A(7), B(4), F(12), G(14)
- E: B(11), C(10), F(15), I(3)
- F: B(8), D(12), E(15), G(17), H(13), I(3)
- G: A(9), D(14), F(17), H(18)
- H: F(13), G(18), I(2)
- I: C(1), E(3), F(3), H(2)

Soluzione del esercizio 1

- la foresta di scoperta e un possibile ciclo di lunghezza 7:



- gli archi in questione
  - 6/13 → 16/23: non poù esistere perché se ci fosse allora 16/23 dovrebbe essere scoperto tramite questo arco
  - 7/8 → 3/4: poù esistere perché la visita di 3/4 è già finita (il vertice è nero) quando la visita di 7/8 comincia (il vertice diventa grigio)
  - 16/23 → 6/13: come il precedente
  - 17/18 → 15/24: poù esistere perché la visita di 15/24 è già cominciata (il vertice è grigio) quando la visita di 17/18 comincia (il vertice diventa grigio)

Soluzione del esercizio 2

La dimostrazione poù essere fatta per costruzione, cioè descriviamo un algoritmo che trova un ciclo in un grafo in cui ogni nodo ha almeno un arco entrante.

Partiamo da un nodo qualsiasi del grafo. Prendiamo un arco qualsiasi che entra in questo nodo (deve esserci almeno uno). Passiamo al nodo da cui questo arco esce. In questo nodo scegliamo di nuovo un arco entrante (deve essereci almeno uno) e passiamo al nodo da cui questo arco esce. Ripetendo questo passo, sicuramente torniamo in un nodo in cui siamo già stati. Questo vuole dire che abbiamo percorso un ciclo in senso contrario, cioè abbiamo trovato un ciclo.

Soluzione del esercizio 3

Per avere un algoritmo che costruisce un albero di copertura massimo invece di minimo basta considerare gli archi del grafo in ordine non crescente di peso invece del ordine non decrescente:

MST_Kruskal(G)
  $A \leftarrow \emptyset$
  for $\forall v \in V$ **do**
    Make_set(v)
  ordina gli archi in ordine non crescente di peso
  for $\forall (u, v) \in E$ nell'ordine **do**
    if Find(u)$\neq$Find(v) **then**
      $A \leftarrow A \cup (u, v)$
      Union(u, v)

Applicando l'algoritmo, vengono scelti i seguenti archi e uniti i seguenti insieme di nodi:

- arco scelto GH di peso 18, insiemi uniti: {G} e {H}

- arco scelto GF di peso 17, insiemi uniti: {G,H} e {F}
- arco scelto EF di peso 15, insiemi uniti: {G,H,F} e {E}
- arco scelto GD di peso 14, insiemi uniti: {G,H,F,E} e {D}
- arco scelto BE di peso 11, insiemi uniti: {G,H,F,E,D} e {B}
- arco scelto CE di peso 10, insiemi uniti: {G,H,F,E,D,B} e {C}
- arco scelto AG di peso 9, insiemi uniti: {G,H,F,E,D,B,C} e {A}
- arco scelto IE di peso 3, insiemi uniti: {G,H,F,E,D,B,C,A} e {I}