Analisi e progettazione di algoritmi

(III anno Laurea Triennale - a.a. 2022/23)

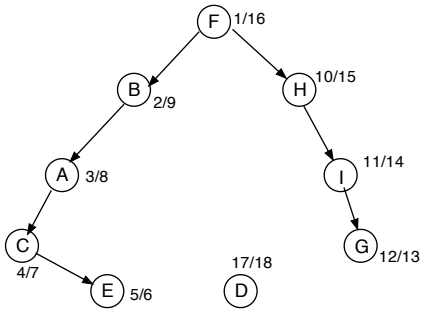
Prova scritta 19 giugno 2023

Esercizio 1 Un grafo orientato G ha nodi A, B, C, D, E, F, G, H, I. Non ne conosciamo gli archi, ma sappiamo che la sua visita in profondità produce la seguente sequenza di inizio/fine visita: A 3/8, B 2/9, C 4/7, D 17/18, E 5/6, F 1/16, G 12/13, H 10/15, I 11/14.

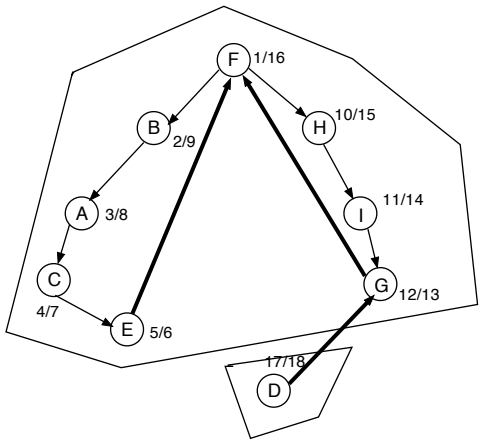
1. Si disegni la foresta DFS ottenuta attraverso la visita.
2. Si disegni G, aggiungendo archi alla foresta DFS, in modo che sia formato da due componenti fortemente connesse. Se non è possibile si spieghi perché.
3. Si disegni G, aggiungendo archi alla foresta DFS, in modo che sia formato da una sola componente fortemente connessa. Se non è possibile si spieghi perché.

Soluzione

1. La foresta DFS ottenuta attraverso la visita è la seguente:



2. Un G formato da due componenti fortemente connesse è il seguente:



3. Non è possibile che G sia formato da una sola componente fortemente connessa, perché dovrebbe esserci un arco che collega un altro nodo a D, ma in tal caso la visita di D inizierebbe durante la visita di questo altro nodo, in disaccordo con i timestamp dati.

Esercizio 2 Si consideri un array $A[0..n-1]$ di interi, ordinato in senso non decrescente.

1. Si scriva un algoritmo divide et impera per decidere se l'array contiene oppure no elementi duplicati. Suggerimento: ispirarsi alla ricerca binaria ricorsiva (Esempio 1.10 delle note).
2. Giustificare la correttezza dell'algoritmo proposto.
3. Analizzare il costo computazionale dell'algoritmo proposto.
4. Potrebbe esistere un algoritmo asintoticamente più efficiente?

Soluzione

1. Un algoritmo divide et impera che decide se l'array contiene oppure no elementi duplicati è il seguente:

```
dup(A)
  dup(A,0,n-1)

dup(A,inf,sup)
  if inf ≥ sup return false
  mid = (inf+sup)/2
  if (A[mid]=A[mid+1]) return true
  return dup(A,inf,mid) or dup(A,mid+1,sup)
```

2. La correttezza dell'algoritmo può essere provata per induzione aritmetica forte sul numero di elementi. Nel caso di nessun elemento o un elemento solo l'algoritmo restituisce falso correttamente. Nel caso di almeno due elementi, la porzione di array viene divisa in due parti. Se l'ultimo elemento della prima parte è uguale al primo della seconda l'array contiene duplicati, e l'algoritmo restituisce vero correttamente. Altrimenti, dato che l'array è ordinato, vi sono duplicati solo se una delle due parti contiene duplicati, come correttamente calcolato dalle due chiamate ricorsive per ipotesi induttiva.
3. La relazione di ricorrenza è la seguente (conviene esprimere n come 2^k):

   $T(2^0) = 1$
   $T(2^k) = 1 + 2 \cdot T(2^{k-1})$, per $k > 0$.

   Utilizzando la tecnica per sostituzioni successive si ha:

   $$T(2^k) = 1 + 2 \cdot T(2^{k-1}) = 1 + 2 + 2^2 \cdot T(2^{k-2}) = \ldots = 1 + 2 + \ldots + 2^k = \frac{2^{k+1}-1}{2-1}$$

   quindi $T(n) = O(n)$.
4. Non è possibile dare un algoritmo più efficiente, ossia il problema ha complessità $\Omega(n)$, in quanto è sicuramente necessario esaminare tutti gli elementi.[1]

Esercizio 3 Considera una sequenza composta dai primi 10 numeri naturali. Costruisci l'input peggiore per la versione deterministica di *QuickSort* che sceglie come *pivot* l'ultimo elemento. Sapresti trovarne un altro? Nelle notazioni utilizzate a lezione, quanto vale $p_{ij}(1)$?

[1] Si noti che in questo esempio l'algoritmo divide-et-impera non è più efficiente di un algoritmo iterativo che controlla ogni coppia di elementi contigui.

Soluzione Nel caso peggiore la scelta del *pivot* lascia sempre una delle due sottosequenze vuota. Se il *pivot* è l'ultimo elemento l'input peggiore è la sequenza ordinata da 1 a 10 o quella ordinata da 10 a 1. Poiché tutte le coppie sono confrontate $p_{ij}(1)$ è sempre uguale a 1.

Guida alla correzione Esercizio 1: 1.1: 5 punti a chi collega D 1.3: 7 punti a chi ha almeno detto che non si poteva fare
Esercizio 2: max 3 punti su 2.1, 2.2., 2.3 a chi ha risolto un altro problema (trovare se un dato elemento compare due volte) 2.1: 4 a chi ha almeno capito il problema e lo schema ricorsivo; 2.2: 4 a chi almeno menziona correttamente schema di induzione forte; 10 a chi ha capito che il punto chiave è che la sequenza sia ordinata, se no max 8