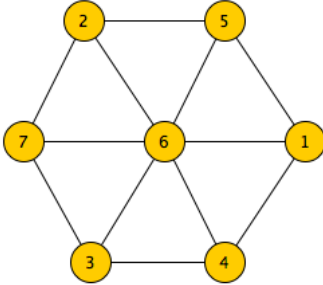


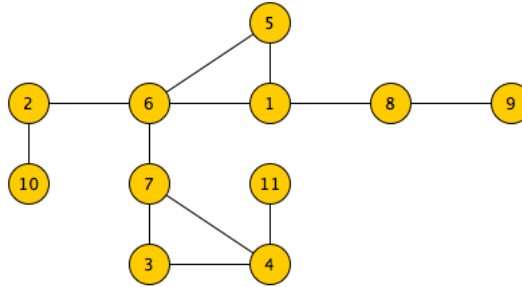
Progettazione di Algoritmi - Prova Intermedia Aprile 2017

Esercizio [DFS in grafi semplici] (max 5) Si consideri il grafo G nella figura qui sotto riportato. Si assuma G rappresentato in memoria tramite liste di adiacenza e in ogni lista i nodi compaiono in ordine crescente.



1. Qual'è il numero minimo di nodi da eliminare dal grafo G perchè il grafo G' risultante abbia almeno un punto di articolazione? **Motivare bene la risposta**
2. Qual'è il numero minimo di archi da eliminare dal grafo G perchè il grafo G' risultante abbia almeno un ponte? **Motivare bene la risposta**

Esercizio [Ponti] (max 5) Si consideri il grafo G nella figura qui sotto riportato. Si assuma G rappresentato in memoria tramite liste di adiacenza e in



ogni lista i nodi compaiono in ordine crescente.

1. Elencare i ponti del grafo G nell'ordine con cui vengono individuati dall'algoritmo visto a lezione e basato sulla visita DFS assumendo che la visita parta dal nodo 1
2. Qual'è il numero minimo di archi che bisogna aggiungere al grafo perchè non abbia più ponti? **Motivare bene la risposta**

Esercizio [vettore dei padri] (max 6) Un grafo aciclico e connesso G viene rappresentato tramite il vettore dei padri P relativo ad una visita DFS di G a partire da un suo nodo arbitrario. Dare lo pseudo-codice di un algoritmo che, preso in input il vettore dei padri P , determina il numero di nodi di grado 1 presenti in G . L'algoritmo deve avere complessità $O(n)$.

Ad esempio: Per il grafo di 5 nodi il cui vettore dei padri P contiene nell'ordine: $2, 4, 2, 4, 1$ l'algoritmo deve produrre il numero 3. Per il grafo di 5 nodi il cui vettore dei padri P contiene nell'ordine: $2, 2, 2, 3, 1$, l'algoritmo deve produrre 2.

Esercizio [Sottostringa comune] (max 10) Si consideri il seguente problema. Data una sequenza di simboli X , sono sue sottosequenze tutte le stringhe che si possono ottenere eliminando 0 o più simboli da X . Ad esempio bab e abz sono sottosequenze di $abacbaz$ mentre bbb , $aaab$ e za non lo sono.

Date due sequenze binarie X e Y vogliamo determinare la lunghezza della più lunga sottosequenza comune ad X e Y . Ad esempio, se $X = 10101$ e $Y = 00111$ allora una sottosequenza comune di lunghezza massima è 011 .

Viene proposto il seguente algoritmo:

```

INPUT due sequenze  $X = x_1, \dots, x_n$  e  $Y = y_1, \dots, y_m$ 
   $NX \leftarrow 0$ 
  FOR  $i \leftarrow 1$  TO  $n$  DO
    IF  $x_i = 0$  THEN  $NX \leftarrow NX + 1$ 
  ENDFOR
   $NY \leftarrow 0$ 
  FOR  $i \leftarrow 1$  TO  $m$  DO
    IF  $y_i = 0$  THEN  $NY \leftarrow NY + 1$ 
  ENDFOR
OUTPUT  $\max\{\min\{NX, NY\}, \min\{n - NX, m - NY\}\}$ 

```

1. Provare che la lunghezza prodotta in output dall' algoritmo è la lunghezza di una sottosequenza comune alle due sequenze di input.
2. Provare che l' algoritmo non trova necessariamente la sottosequenza comune più lunga.
3. Dimostrare che l' algoritmo garantisce un rapporto di approssimazione limitato da 2.

Esercizio [complessità del divide et impera] (max 3) Supponi di dover scegliere tra i seguenti tre algoritmi che risolvono un certo problema:

1. L' algoritmo *A* risolve ricorsivamente il problema su istanze di dimensione n riducendolo in tempo $\Theta(\log^2 n)$ ad un sottoproblema di dimensione $n/4$.
2. L' algoritmo *B* risolve ricorsivamente il problema su istanze di dimensione n dividendolo in tempo $O(1)$ in 2 sottoproblemi ciascuno di dimensione $n/4$ e poi ricombinando le soluzioni in tempo $\Theta(\sqrt{n})$.
3. L' algoritmo *C* risolve il problema su istanze di dimensione n dividendolo in tempo $O(1)$ in 8 sottoproblemi ciascuno di dimensione $n-3$ e poi ricombinando le soluzioni in tempo $\Theta(1)$.

Determinare la complessità asintotica dei tre algoritmi e individuare quello asintoticamente più veloce.

Esercizio [radice] (max 7) Scrivere una procedura che, dato l' intero n , calcola il valore $\lfloor \sqrt{n} \rfloor$ utilizzando $O(\log n)$ operazioni aritmetiche.