

008AA – ALGORITMICA E LABORATORIO
Secondo Compitino, 25 Maggio 2016

Cognome Nome:

N. Matricola:

Corso: A B

Esercizio 1. (10 punti)

Un grafo non orientato è detto **2-colorabile** se è possibile attribuire a ogni vertice uno tra due colori in modo che vertici adiacenti siano colorati con colori distinti. Progettare un algoritmo efficiente per verificare se un grafo G , connesso e non orientato, è 2-colorabile.

Esercizio 2. (8 punti)

Dato un insieme di n elementi contenuti in un array P , l'algoritmo seguente costruisce in P tutte le permutazioni di tali elementi. Il calcolo è avviato con la chiamata iniziale `GeneraPermutazioni(P,1)`.

```
GeneraPermutazioni(P,k)
if (k == n) Elabora(P)
else {
    for i = k to n {
        Scambia P[k] e P[i]
        GeneraPermutazioni(P,k+1)
        Scambia P[k] e P[i]
    }
}
```

Individuare l'ordine in cui vengono costruite le permutazioni, simulando il comportamento dell'algoritmo nel caso in cui P contenga le prime tre lettere dell'alfabeto a , b , c , e la procedura `Elabora(P)` consista nella stampa del contenuto dell'array.

Esercizio 3. (10 punti)

Si consideri un array S di n chiavi intere che possono assumere solo $d \leq n$ valori distinti.

- Si dia il codice di un algoritmo che con un'unica scansione di S conti il numero di chiavi distinte che occorrono un numero dispari di volte in S . Usare un dizionario D inizialmente vuoto (non interessa l'implementazione di D).
- Facendo l'assunzione che D sia implementato come un albero AVL, si analizzi la complessità in funzione di n e del numero d di chiavi distinte.

Esercizio 4. (2 punti)

Dare la complessità dell'algoritmo `GeneraPermutazioni`, specificando se si tratta di un algoritmo polinomiale, pseudopolinomiale oppure esponenziale.