Problem Set 1

docente: Luciano Gualà

Esercizio 1 (*notazione asintotica*)
Siano $f(n), g(n), h(n)$ tre funzioni asintoticamente positive. Inoltre, sia $c > 1$ una costante reale positiva. Si dimostrino o confutino le seguenti affermazioni:

1. $2^{f(n)+2^c} = \Theta(2^{f(n)})$.
2. $g(n) = \Theta(1)$ implica $2^{f(n)+g(n)} = O(2^{f(n)})$.
3. $g(n) = o(f(n))$ implica $2^{f(n)+g(n)} = O(2^{f(n)})$.
4. $f(n) + g(n) + h(n) = \Theta(\max\{f(n), g(n), h(n)\})$.
5. $f(n) = \Theta(\log n)$ implica $\log n^{f(n)} = O(\log^c n^{g(n)})$.
6. $f(n) = \Theta(f(c \cdot n))$.
7. $f(n) = \Theta(f(c + n))$.

Esercizio 2
Sia $A[1 : n]$ un vettore ordinato di n interi distinti. Ogni $A[i]$ può essere positivo, negativo o zero. Progettare un algoritmo con complessità temporale $o(n)$ che trova, se esiste, un indice i tale che $A[i] = i$.

Esercizio 3 Si consideri una sequenza di n pedine allineate in cui ogni pedina è colorata di rosso, di verde o di bianco. Una mossa consiste nello scambiare di posto due pedine adiacenti della sequenza. Si vogliono riposizionare le pedine in modo che i loro colori formino la bandiara italiana (cioè compaiano nell'ordine, da sinistra a destra, prima tutte le pedine verdi, poi le bianche e infine le rosse).

Progettare un algoritmo che risolve il problema con un numero asintoticamente ottimo di scambi. Dimostrarne l'ottimalità.

Esercizio 4 (*problemi algoritmici, puzzle e modelli di calcolo.*) Si consideri una sequenza di n pedine allineate in cui ogni pedina è colorata di rosso, di verde o di bianco. Una mossa consiste nello scambiare di posto due pedine della sequenza (non necessariamente adiacenti). Si vogliono riposizionare le pedine in modo che i loro colori formino la bandiara italiana (cioè compaiano nell'ordine, da sinistra a destra, prima tutte le pedine verdi, poi le bianche e infine le rosse).

- Progettare un algoritmo che risolve il problema effettuando al più $O(n)$ scambi.
- Si noti che, nella precedente versione, si sta implicitamente assumendo un modello di calcolo in cui le uniche operazioni che hanno un costo sono le operaioni di scambio. In particolare, non è conteggiato il costo per individuare le pedine da scambiare. Si supponga adesso che le pedine siano elementi di un vettore $A[1 : n]$ e si progetti un algoritmo con complessità temporale $O(n)$ e complessità spaziale $O(1)$. Anche in questo caso l'unica mossa ammissibile è quella di scambiare due elementi del vettore, ma il costo computazionale dell'algoritmo è misurato come numero (asintotico) di operazione elementari nel modello RAM (a costi uniformi).

Esercizio 5 *(Uno strano algoritmo di ordinamento)*
L'algoritmo ricorsivo `gualasort` è un algoritmo di ordinamento ricorsivo, la cui chiamata iniziale `gualasort`$(V, 1, n)$ ordina il vettore $V[1; n]$ di n numeri su cui è chiamato. O almeno si spera. Dimostrate la correttezza dell'algoritmo e discutetene la sua complessità computazionale.

Algorithm 1: `gualasort`(V, i, j)

$N \leftarrow j - i + 1$;
if $N = 2$ **then**
  if $V[i] > V[i+1]$ **then**
    scambia $V[i]$ e $V[i+1]$
if $N > 2$ **then**
  $m_1 = i + \lfloor \frac{N}{3} \rfloor$;
  $m_2 = i - 1 + \lceil \frac{2}{3} N \rceil$;
  `gualasort`(V, i, m_2);
  `gualasort`(V, m_1, j);
  `gualasort`(V, i, m_2);