Soluzione Testo del 22/6/18

Esercizio

```
Secondo (A, i, J)    <---- invocato con i = 1, J = n
   if (J < i) return <-∞, -∞>
   if (i == J) return <A[i], -∞>
   m = (i+J)/2;
   <p1, s1> = Secondo (A, i, m);
   <p2, s2> = Secondo (A, m+1, J);
   p = max (p1, p2);
   if (p == p1) s = max (s1, p2);
         else s = max (s2, p1);
   return <p, s>;
```

La procedura so descritta restituisce il primo e il secondo elemento dell'array A nella porzione [i, J]. Eventualmente un valore è $-\infty$ se invocato su meno di 2 elementi. Pertanto s sarà la risposta al problema.

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + O(1) & n \geq 2 \\ O(1) & n < 2 \end{cases}$$

La cui soluzione è $O(n)$, utilizzando p.e. il Teorema principale.

Esercizio 2

$$T(n) = \begin{cases} a\,T\left(\frac{n}{2}\right) + O(n^2) & n > 1 \\ O(1) & n \leq 1 \end{cases}$$

$$n^{\lg_2 a} \sim n^2 \longleftrightarrow \lg_2 a = 2 \longleftrightarrow a = 4$$

I casi pertanto sono 3:

Caso 1 : $n^2 = O\left(n^{\lg_2 a - \varepsilon}\right) \longleftrightarrow a > 4$

Caso 2 : $n^2 = \Theta\left(n^{\lg_2 a}\right) \longleftrightarrow a = 4$

Caso 3 : $n^2 = \Omega\left(n^{\lg_2 a + \varepsilon}\right) \longleftrightarrow a < 4$

$a\,f\left(\frac{n}{2}\right) \leq c\,f(n)$ condizione di regolarità

$a\left(\frac{n}{2}\right)^2 \leq c\,n^2 \Longleftrightarrow a\frac{\cancel{n^2}}{4} \leq c\,\cancel{n^2}$

basta prendere $\frac{a}{4} \leq c < 1$ che è

possibile in quanto $\boxed{a < 4}$

La soluzione è dunque ottenuta per casi:

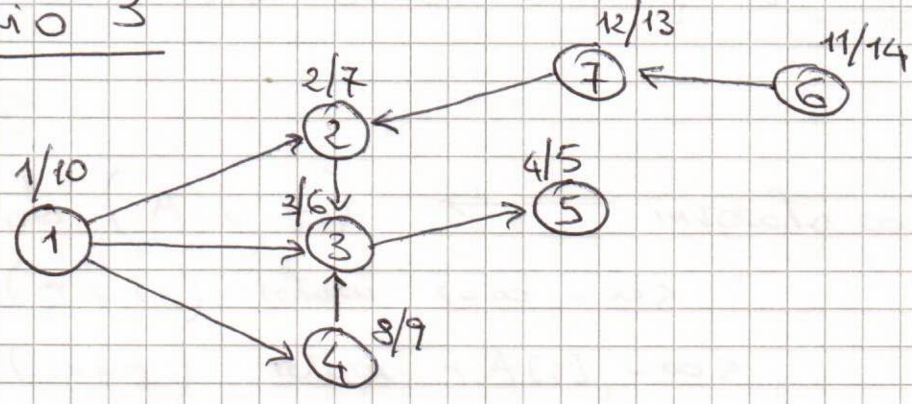
$a > 4 \longrightarrow \Theta\left(n^{\lg_2 a}\right)$

$a = 4 \longrightarrow \Theta\left(n^2 \lg n\right)$

$a < 4 \longrightarrow \Theta\left(n^2\right)$

Per $a \leq 8$ si ha che $T(n) = O(n^3)$

Esercizio 3



Si esegue una DFS partendo dal nodo 1 e si assegnano le marche temporali di inizio/fine della visita di ogni nodo.

Dopo di che si ordinano i vertici per tempo di fine decrescente, e quindi si ottiene la sequenza

6, 7, 1, 4, 2, 3, 5

Esercizio 4

Si veda il libro di testo.

Esercizio 1 (soluzione bis)

In classe è stata presentata la funzione che calcola ricorsivamente il massimo di un vettore. Sia questa MAX (A, i, J) e si assume che ritorni l'indice del massimo tra le posizioni i e J.

```
Secondo (A)    // restituisce la posizione del secondo.
   m = Max (A, 1, n)
   x = Max (A, 1, m-1);
   y = Max (A, m+1, n);
   if (x<y) return y else return x;
```