Esercizio 1 Presso un'azienda sono disponibili per le prossime n settimane compiti che possono essere suddivisi in *semplici* o *impegnativi*. Il guadagno che si ricava dall'eseguire i compiti *semplici* della j-esima settimana è x_j, $1 \le j \le n$, mentre quello che deriva dall'eseguire i compiti *impegnativi* è y_j, $1 \le j \le n$. Bisogna pianificare che tipo di compiti svolgere nelle prossime n settimane in modo da massimizzare il ricavo tenendo conto che a ciascuna settimana in cui si è scelto di svolgere i compiti *impegnativi* deve seguire necessariamente una settimana di riposo.

Ad esempio: per

X=	10	1	10	10
Y=	5	50	5	1

- La pianificazione $[x, x, x, x]$ che prevede di scegliere in ciascuna settimana i compiti *semplici* ha valore $10 + 1 + 10 + 10 = 31$.
- La pianificazione $[y, *, y, *]$ che prevede di scegliere sempre i compiti *impegnativi* ha valore $5+5 = 10$
- La pianificazione che massimizza il ricavo è la $[x, y, *, x]$ che vale $10 + 50 + 10 = 70$ e prevede di scegliere i compiti impegnativi nella sola seconda settimana.

Per risolvere il problema si propone il seguente algoritmo greedy

INPUT: le tabelle dei ricavi $X = \{x_j | 1 \le j \le n\}$ e $Y = \{y_j | 1 \le j \le n\}$
  $j \leftarrow 1$
  WHILE $j < n$ DO
    IF $y_j > x_j + x_{j+1}$ THEN
      $SOL[j] \leftarrow y$
      $SOL[j+1] \leftarrow *$
      $j \leftarrow j + 2$
    ELSE
      $SOL[j] \leftarrow x$
      $j \leftarrow j + 1$
    ENDIF
  ENDWHILE
  IF $j = n$ AND $x_n \ge y_n$ THEN $SOL[n] = x$
  IF $j = n$ AND $y_n > x_n$ THEN $SOL[n] = y$
OUTPUT SOL

1. *(max 10 punti)* provare che l'algoritmo proposto non è corretto fornendo un'istanza per cui la pianificazione SOL prodotta non è quella a ricavo massimo.
2. *(max 10 punti)* proporre un algoritmo che in tempo $O(n)$ calcola il valore della pianificazione a ricavo massimo (*Suggerimento: utilizzare la programmazione dinamica calcolando i* $3n$ *valori* $M[x, j]$, $M[y, j]$ *e* $M[*, j]$ *con* $1 \le j \le n$ *dove* $M[x, j]$ *è il valore della pianificazione a ricavo massimo per le prime* j *settimane quando nella* j*-esima si sceglie di eseguire i compiti* semplici, $M[y, j]$ *quella in cui nella* j*-esima si sceglie di eseguire i compiti* difficili *ed* $M[*, j]$ *quella in cui nella* j*-esima si è a riposo.*)
3. *(max 10 punti)* modificare l'algoritmo proposto al punto 2 in modo da avere in output una pianificazione di ricavo massimo anziché il suo valore. La modifica deve avere un costo addittivo $O(n)$.

Soluzione Esercizio 1

1. Si consideri l'istanza: $X =$

X=	1	1
Y=	3	100

   L'algoritmo produce la pianificazione $[y, *]$ di valore 3 mentre la pianificazione di valore massimo è $[x, y]$ che ha valore $1 + 100 = 101$.

2. Ovviamente per $j = 1$ si ha $M[x, 1] = x_1$, $M[y, 1] = y_1$ e $M[*, 1] = 0$.

   Per $j > 2$ deve aversi:

   - $M[x, j] = \max\{M[x, j-1], M[*, y-1]\} + x_j$
   - $M[y, j] = \max\{M[x, j-1], M[*, y-1]\} + y_j$
   - $M[*, j] = \max\{M[x, j-1], M[y, j-1], M[*, j-1]\}$

   Pertanto, tenendo conto che il valore della pianificazione di ricavo massimo è dato da $\max\{M[x, n], M[y, n], M[*, n]\}$, l'algoritmo per calcolarlo è:

   INPUT: le tabelle dei ricavi $X = \{x_j | 1 \leq j \leq n\}$ e $Y = \{y_j | 1 \leq j \leq n\}$
   $M[x, 1] \leftarrow x_1$
   $M[y, 1] \leftarrow y_1$
   $M[*, 1] \leftarrow 0$
   FOR $j \leftarrow 2$ TO n DO
   $M[x, j] \leftarrow M[x, j-1] + x_j$
   IF $M[x, j-1] < M[*, j-1]$ THEN $M[x, j] \leftarrow M[*, j-1] + x_j$
   $M[y, j] \leftarrow M[y, j-1] + y_j$
   IF $M[x, j-1] < M[*, j-1]$ THEN $M[y, j] \leftarrow M[*, j-1] + y_j$
   $M[*, j] \leftarrow M[*, j-1]$
   IF $M[*, j] < M[y, j-1]$ THEN $M[*, j] \leftarrow M[y, j-1]$
   IF $M[*, j] < M[*, j-1]$ THEN $M[*, j] \leftarrow M[*, j-1]$
   ENDFOR
   $SOL \leftarrow M[x, n]$
   IF $SOL < M[y, n]$ THEN $SOL \leftarrow M[y, n]$
   IF $SOL > M[*, n]$ THEN $SOL \leftarrow M[*, n]$
   OUTPUT SOL

3. Per ottenere la pianificazione di ricavo massimo basta ripercorrere all'indietro le scelte che hanno portato a calcolarne il valore sostituendo quindi le ultime 4 righe di codice dell'algoritmo precedente con quelle che seguono

   $SOL[n] \leftarrow x$
   IF $M[SOL[n], n] < M[y, n]$ THEN $SOL[n] \leftarrow y$
   IF $M[SOL[n], n] < M[*, n]$ THEN $SOL[n] \leftarrow *$
   $j \leftarrow n - 1$
   WHILE $j > 0$ DO
   $tipo \leftarrow SOL[j+1]$
   IF $tipo = *$ THEN $SOL[j] \leftarrow y$
   ELSE
   IF $M[x, j] + tipo_{j+1} = M[tipo, j+1]$ THEN $SOL[j] = x$
   ELSE $SOL[j] = *$
   $j \leftarrow j - 1$
   ENDWHILE
   OUTPUT SOL