

PROVA INTERMEDIA DEL 27 NOVEMBRE 2008

Esercizio 1. Si consideri la struttura dati *max-heap* vista in classe (heap binario in cui il massimo è nella radice). Definire una corrispondente struttura dati *min-heap* in cui la radice contiene l'elemento con chiave minima e la proprietà di ordinamento a heap è invertita:

$$\text{chiave}(\text{parent}(v)) \leq \text{chiave}(v)$$

per ogni nodo v differente dalla radice. Riscrivere le procedure `fixHeap` e `heapify` nel caso di min-heap. Discutere la correttezza dell'algorithmo su cui si basa la procedura `heapify`. Infine applicare `heapify` al seguente array mostrandone l'evoluzione:

$$H = 81 \ 10 \ 78 \ 55 \ 44 \ 70 \ 60 \ 6 \ 31 \ 20$$

Esercizio 2. Sia X un array di dimensione n . Considerare gli algoritmi `mistero1` e `mistero2` mostrati di seguito:

algoritmo mistero1 (*array* X , *indici* i e f) \rightarrow *valore*

1. **if** ($i = f$) **then return** $X[i]$
2. **else**
3. $v \leftarrow \text{mistero1}(X, i + 1, f)$
4. **if** ($X[i] < v$) **then return** v
5. **else return** $X[i]$

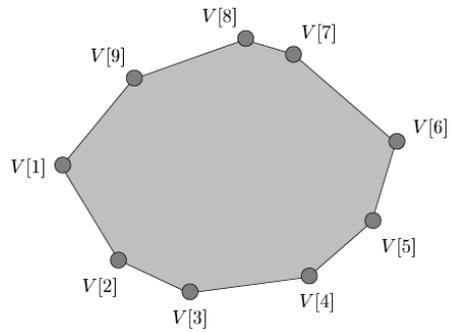
algoritmo mistero2 (*array* X , *indici* i e f) \rightarrow *valore*

1. **if** ($i = f$) **then return** $X[i]$
2. **else**
3. $m \leftarrow (i + f)/2$
4. $v1 \leftarrow \text{mistero2}(X, i, m)$
5. $v2 \leftarrow \text{mistero2}(X, m + 1, f)$
6. **if** ($v1 < v2$) **then return** $v2$
7. **else return** $v1$

In entrambi i casi, gli indici passati nella chiamata iniziale sono $i = 1$ e $f = n$.

1. I due algoritmi risolvono lo stesso problema? Se si, quale? Motivare la risposta.
2. Uno dei due algoritmi è più efficiente dell'altro da un punto di vista asintotico? Motivare la risposta analizzando il tempo di esecuzione di entrambi.

Esercizio 3. In un poligono convesso tutti gli angoli sono minori di 180° e non ci sono intersezioni tra spigoli se non ai loro estremi. Un esempio di poligono convesso è mostrato in figura.



Rappresentiamo un poligono convesso come un array $V[1..n]$ tale che:

1. ciascun elemento dell'array rappresenta un vertice del poligono tramite un coppia di coordinate (x, y) ;
2. $V[1]$ è il vertice con coordinata x minima;
3. i vertici appaiono in ordine antiorario.

Per semplicità, si può assumere che le coordinate x e y di tutti i vertici siano distinte. Progettare un algoritmo che trovi il punto con coordinata y massima in tempo $O(\log n)$ nel caso peggiore. Discutere correttezza e tempo di esecuzione dell'algoritmo proposto.

| | |
|------|---------|
| NOME | COGNOME |
|------|---------|

Soluzione Esercizio 1:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

PROVA INTERMEDIA DEL 27 NOVEMBRE 2008

| | |
|------|---------|
| NOME | COGNOME |
|------|---------|

Soluzione Esercizio 2:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

PROVA INTERMEDIA DEL 27 NOVEMBRE 2008

| | |
|------|---------|
| NOME | COGNOME |
|------|---------|

Soluzione Esercizio 3:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

PROVA INTERMEDIA DEL 27 NOVEMBRE 2008