Possibile soluzione al Problema 1

Fabrizio d'Amore

Esame FI2 del 19 gennaio 2012

Problema 1

Testo problema 1

Si considerino i seguenti metodi:

```
public static int[] cip(int[] v, int n) {
      if(n < v.length) return cucu(v);
      v = ciop(v);
      v = ciop(v);
      return cip (v, n);
}

private static int[] ciop(int[] v) {
      int[] w = new int[2*v.length];
      int i = 0, j = v.length - 1, k = 0;
      while(i < v.length) {
              w[k++] = v[i++];
              w[k++] = v[j--];
      }
      return w;
}

private static int[] cucu(int[] v) {
      int[] w = new int[1];
      w[0] = cucu(v, 0);
      return w;
}

private static int cucu(int[] v, int i) {
      if(i >= v.length) return 0;
      int r = 0;
      if(i < v.length) r += v[i];
      return r + cucu(v, i+1);
}
```

(a) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `cip(int[] v, int n)` in funzione della dimensione dell'array `v` (indicata con `|v|`) e del valore `n`.

(b) Esprimere il costo calcolato al punto precedente in funzione della dimensione dell'input (indicata con x).

(c) Cosa restituisce `cip(a, 10)` quando l'array `a` ha solo una cella, contenente il valore `1`?

Possibile svolgimento

Punto (a)

Notiamo innanzi tutto che il metodo `cip` è ricorsivo: nel suo passo base fa uso del metodo `cucu`, in quello ricorsivo fa uso di `ciop` (due volte). Le altri fonti di costo di `cip` contribuiscono con un costo *costante*. Determiniamo dapprima i costi $\mathcal{C}_{\texttt{cucu}}$ e $\mathcal{C}_{\texttt{ciop}}$, rispettivamente dei metodi `cucu` e `ciop`.

Il metodo `cucu(int[] v, int i)` è un metodo ricorsivo (ricorsione lineare di coda) che viene attivato `v.length - i` volte; tutte le sue istruzioni hanno banalmente costo $\Theta(1)$. Poiché `cucu(int[] v)` chiama `cucu(v, 0)`, se ne deriva immediatamente che $\mathcal{C}_{\texttt{cucu}}(m) = \Theta(m)$.

Il metodo `ciop(int[] v)` esegue m iterazioni, in ciascuna della quale paga un costo costante. Notare che restituisce un array di dimensione doppia rispetto quella dell'input, riempito con i valori dell'array di input, ciascuno presente due volte (non interessa in questa sede in quale ordine). Risulta dunque $\mathcal{C}_{\texttt{ciop}}(m) = \Theta(m)$.

Per valutare il costo $\mathcal{C}_{\texttt{cip}}(m, n)$, ove $m = \texttt{v.length}$ e $n = \texttt{n}$, supponiamo dapprima $m \leq n$ e calcoliamo il numero di attivazioni ricorsive di `cip`. Durante l'esecuzione di `cip` l'array `v` viene raddoppiato due volte, dunque la successiva attivazione ricorsiva vede un input di dimensione quadruplicata. Per determinare il numero totale di attivazione ricorsive occorre dunque calcolare quante volte occorre quadruplicare m per superare n. In formule, il numero di attivazioni di `cip` è pari a $1 + k$ (chiamata iniziale, più tutte le chiamate ricorsive), essendo k il minimo intero tale che $4^k m > n$, cioè $k = \lceil \log_4(n/m) \rceil$.

Ciascuna esecuzione di `cip` comporta un costo, non considerando i costi delle successive ricorsioni, pari a $\Theta(m)$, facendo però attenzione al fatto che, ad ogni chiamata, m risulta quadruplicato. Il costo di tutti questi contributi sarà dato dunque dalla somma di $1 + \lceil \log_4(n/m) \rceil$ componenti di costo pari a $\Theta(4^i m)$, per $i = 0, 1, 2, \ldots, \lceil \log_4(n/m) \rceil$. In formule:[1]

$$\begin{aligned}
\sum_{i=0}^{\lceil \log_4(n/m) \rceil} \Theta(4^i m) &= \Theta(\sum_{i=0}^{\lceil \log_4(n/m) \rceil} (4^i m)) = \Theta(m \sum_{i=0}^{\lceil \log_4(n/m) \rceil} 4^i) = \Theta(m \frac{4^{\lceil \log_4(n/m) \rceil + 1} - 1}{4 - 1}) = \\
&= \Theta(m \frac{4^{\log_4(n/m) + 1} - 1}{3}) = \Theta(m \frac{4n/m - 1}{3}) = \Theta(\frac{4n - m}{3}) = \Theta(n)
\end{aligned}$$

ricordando che $n \geq m$. Il costo è pertanto $\Theta(n) + \mathcal{C}_{\texttt{cucu}}(m')$ (costo del passo base), avendo indicato con m' la dimensione finale dell'array: $m' = 4^{\lceil \log_4(n/m) \rceil} m = \Theta(4^{\log_4(n/m)} m) = \Theta(n)$. Essendo $m' = \Theta(n)$, e data la linearità di $\mathcal{C}_{\texttt{cucu}}$, risulta $\mathcal{C}_{\texttt{cucu}}(m') = \Theta(n)$. In definitiva, nel caso $n \geq m$, abbiamo $\mathcal{C}_{\texttt{cip}}(m, n) = \Theta(n)$.

Nel caso in cui $m > n$ il metodo `cip` non esegue chiamate ricorsive e termina chiamando direttamente `cucu`, per un costo totale pari a quello di `cucu`: $\Theta(m)$.

Risulta pertanto $\mathcal{C}_{\texttt{cip}}(m, n) = \Theta(\max\{m, n\}) = \Theta(m + n)$.

Punto (b)

La dimensione b dell'input di `cip` è $b = |\texttt{v}| + |\texttt{n}|$, che possiamo scrivere come $b = b' + b''$, essendo $b' = c \cdot m$, per una opportuna costante $c > 1$, e $b'' = \lfloor \log_2 n \rfloor + 1$; possiamo anche dire che l'input ha dimensione $b = \Theta(m + \log_2 n)$. Il punto (b) richiede di esprimere il valore di $\mathcal{C}_{\texttt{cip}}$ in funzione di b.

Possiamo facilmente scrivere che $\mathcal{C}_{\texttt{cip}}(b) = \Theta(b' + 2^{b''})$. Facciamo ora il seguente ragionamento, senza mai dimenticare che siamo interessati al costo asintotico di caso peggiore. Se la dimensione dell'input crescesse di p bit potremmo immaginare che parte di questi p bit vadano a confluire in b' e la restante parte in b''. I due casi estremi sono: tutti i p bit confluiscono in b' e tutti i p bit

[1] In quanto segue, il simbolo Θ è utilizzato impropriamente. Infatti, $\Theta(f(n))$ denota l'insieme delle funzioni che sono $O(f(n))$ e $\Omega(f(n)$: $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$; il simbolo Θ denota pertanto un insieme. In questo documento estendiamo la notazione indicando ancora con Θ un qualsiasi elemento dell'insieme rappresentato.

confluiscono in b''. Nel primo caso il costo passerebbe da $\Theta(b' + 2^{b''}$ a $\Theta(b' + p + 2^{b''})$, nel secondo arriverebbe invece a $\Theta(b' + 2^{b''+p})$; ovviamente, nel secondo caso, l'incremento di costo sarebbe (assai) superiore. Nei casi intermedi, ove p si ripartisce fra b' e b'', si osserverebbero incrementi inferiori.

Questa osservazione ci porta a dire che gli incrementi di b'' sono dominanti e caratterizzano asintoticamente il costo del metodo. In altre parole, nel caso peggiore, tutti gli incrementi della dimensione dell'input interessano l'intero n. Questo ci porta a concludere che il termine b' è asintoticamnte dominato dall'altro termine. In definitiva: $\mathcal{C}_{\texttt{cip}}(b) = \Theta(2^b)$.

Punto (c)

Sulla base delle considerazioni svolte nel punto (a), possiamo riportare nello schema che segue lo stato dell'array in input, per ciascuna attivazione di `cip`.

# attivazione	array `v`	`cucu(v)`
1	`1`	1
2	`1 1 1 1`	4
3	`1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1`	16

Pertanto, risulta `cip({1}, 10)`=`16`.