

Siano $f(n)$ e $g(n)$ due funzioni non negative e monotone non decrescenti. Dire quali delle seguenti affermazioni sono vere, fornendo un controesempio qualora non lo fossero:

1. $(\min(f(n), g(n)) + \max(f(n), g(n))) \in \Theta(2f(n) + 3g(n))$;
2. $\Omega(|f(n) - g(n)|) \cap O(f(n) + g(n)) = \Theta(f(n) + g(n))$;
3. $f(n) \cdot f(n+2) \in O(f(n+1) \cdot f(n+1))$;
4. $4^{f(n)} + 4^{g(n)} \in O(2^{f(n)+g(n)})$

Le risposte, punto per punto, sono le seguenti:

1. Notare che $(\min(f(n), g(n)) + \max(f(n), g(n))) = f(n) + g(n)$. Abbiamo poi

$$\frac{1}{3}(2f(n) + 3g(n)) \leq f(n) + g(n) \leq \frac{1}{2}(2f(n) + 3g(n)),$$

quindi l'affermazione è vera e le costanti nascoste dalla notazione theta sono $\frac{1}{3}$ e $\frac{1}{2}$.

2. Falso, come si vede prendendo $g(n) = f(n)$, dove $f(n)$ è una qualsiasi funzione non costante. Siccome allora $\Omega(|f(n) - g(n)|)$ è l'insieme di tutte le funzioni definitivamente non negative (cioè, di tutte le funzioni alle quali le nostre notazioni si applicano), il membro sinistro è semplicemente $O(f(n) + g(n))$, che sappiamo essere diverso da $\Theta(f(n) + g(n))$. Concretamente, prendendo $f(n) = g(n) = n^2$, allora $h(n) = n$ appartiene al membro sinistro ma non a quello destro.

L'uguaglianza sussiste invece con l'ipotesi aggiuntiva $g(n) = o(f(n))$.

3. Con una traslazione di indice, possiamo mettere il quesito nella forma $f(n-1) \cdot f(n+1) \in O(f(n) \cdot f(n))$;
Qualche tentativo: prima con una funzione polinomiale, e.g. $f(n) = n^k$. Abbiamo

$$\frac{(n-1)^k (n+1)^k}{n^k n^k} = \left(\frac{n^2 - 1}{n^2} \right)^k < 1$$

per cui la proposizione è in questo caso vera.

Indi con una funzione esponenziale: $f(n) = k^n$

$$\frac{k^{n-1} k^{n+1}}{k^n k^n} = 1.$$

per cui la proposizione è ancora vera.

Anche con $f(n) = k^{n^2}$

$$\frac{k^{(n-1)^2} k^{(n+1)^2}}{k^{n^2} k^{n^2}} = k^{(n-1)^2 + (n+1)^2 - 2n^2} = k^2$$

per cui la proposizione è ancora vera.

Con $f(n) = k^{n^3}$, tuttavia, le cose cambiano: ad es. con $f(n) = k^{n^3}$ si ha

$$\frac{k^{(n-1)^3} k^{(n+1)^3}}{k^{n^3} k^{n^3}} = k^{(n-1)^3 + (n+1)^3 - 2n^3} = k^{6n},$$

rapporto che tende a infinito, per cui la proposizione è falsa.

4. Falso, come si vede prendendo ad esempio $f(n) = 1$ (ricordando che deve essere positiva) e $g(n) = n$. Notare che basterebbe anche prendere una $g(n)$ crescente in modo molto più debole, come $g(n) = \log n$.

Esercizio 2

Il semifattoriale di un intero positivo n è il prodotto di tutti gli interi positivi non maggiori di n , aventi la stessa parità di n . Ad esempio, $5!! = 5 \cdot 3 \cdot 1$; $6!! = 6 \cdot 4 \cdot 2$.

Trovare una semplice $f(n)$ per cui si abbia $\log(n!!) \in \Theta(f(n))$. Dimostrare il risultato trovato, senza dare per buoni i fatti visti a lezione riguardo a $\log(n!)$.

Si può ricostruire, per esempio attraverso il disegno con l'integrale, il risultato visto a lezione, trovando quindi che

$$n \ln n - n \leq \ln(n!) \leq (n+1) \ln(n+1).$$

A questo, punto, distinguiamo due casi a seconda della parità di n .

- se $n = 2k$, allora

$$(2k)! = 2^k k!$$

da cui,

$$(k \ln k - k) + k \leq \ln((2k)!) \leq (k+1) \ln(k+1) + k$$

o più semplicemente

$$k \ln k \leq \ln((2k)!) \leq (k+1)[1 + \ln(k+1)]$$

da cui, tornando a n ,

$$\frac{n}{2} [\ln n - 1] \leq \ln n! \leq (n/2 + 1)[1 + \ln(n/2 + 1)]$$

- se $n = 2k + 1$, allora possiamo sfruttare il fatto che $(2k + 1)!! = (2k + 1)! / (2k)!!$, ed usare il punto precedente. Più sbrigativo è ricorrere al punto precedente notando semplicemente che $(2k)!! < (2k + 1)!! < (2k + 2)!!$.

L'esercizio si poteva anche risolvere più direttamente notando che

$$\int_1^2 \ln(2x) dx \leq \ln(2k)! \leq \int_2^{k+1} \ln(2x) dx$$

e che

$$\int_2^2 \ln(2x + 1) dx \leq \ln(2k + 1)! \leq \int_2^{k+1} \ln(2x + 1) dx.$$

Esercizio 3

Mettere le seguenti funzioni in ordine di crescita non decrescente, segnalando le funzioni che avessero eventualmente lo stesso ordine di crescita:

$$f_1(n) = \binom{6n}{2n}, f_2(n) = \binom{6n}{3n}, f_3(n) = \binom{6n}{4n}, f_4(n) = 8^{2n}, f_5(n) = 2^{8n}.$$

È concesso usare, se necessario, la formula di Stirling.

Per le proprietà di simmetria dei coefficienti binomiali, intanto, notiamo che $f_1(n) = f_3(n)$. Siccome poi i coefficienti binomiali crescono verso il centro, $f_2(n)$ è maggiore di entrambe le precedenti funzioni; ricordiamo che esso è il massimo tra i coefficienti della riga $6n$ del triangolo di Tartaglia.

Ricordiamo poi che i coefficienti di detta riga sommano a 2^{6n} , che è precisamente $f_4(n)$, la quale cresce ovviamente meno di $f_5(n)$. Con considerazioni istantanee siamo dunque arrivati a dire che

$$f_1(n) = f_3(n) \leq f_2(n) \leq f_4(n) < f_5(n).$$

Servono due ulteriori indagini riguardo ai segni \leq per stabilire se essi siano dei minori stretti o delle uguaglianze. Notiamo che

$$\begin{aligned} \frac{f_1(n)}{f_2(n)} &= \frac{\frac{6n!}{2n!4n!}}{\frac{6n!}{3n!3n!}} \\ &= \frac{3n!3n!}{2n!4n!} \\ &= \frac{(3n)(3n-1)\dots(2n+1)}{(4n)(4n-1)\dots(3n+1)} \\ &= \frac{3n}{4n} \frac{3n-1}{4n-1} \dots \frac{2n+1}{3n+1} \\ &= \leq \left(\frac{3}{4}\right)^n \end{aligned}$$

rapporto che tende a zero; gli ordini di grandezza sono dunque diversi.

L'altro dubbio si può risolvere usando la formula di Stirling, come da suggerimento. Abbiamo

$$\begin{aligned}\binom{6n}{3n} &= \frac{6n!}{(3n!)^2} = \frac{\sqrt{2\pi 6n} \left(\frac{6n}{e}\right)^{6n}}{2\pi 3n \left(\frac{3n}{e}\right)^{6n}} (1 + \Theta(1/n)) \\ &= \sqrt{\frac{1}{3\pi n}} 2^{6n} (1 + \Theta(1/n))\end{aligned}$$

da cui si vede che anche questo minore è stretto.

N.B.: il teorema limite centrale del calcolo delle probabilità è un'alternativa alla formula di Stirling per dimostrare in maniera molto rapida che $\lim_{n \rightarrow \infty} f_2(n)/f_4(n) = 0$.

Esercizio 4

Sia dato un grafo non orientato con n nodi e m archi. Vogliamo produrre un elenco delle componenti connesse del grafo, disposte in ordine non crescente di dimensione. Supponiamo, ad esempio, che ogni componente connessa sia rappresentata dal nodo di minimo indice presente in essa.

Dare una descrizione di un algoritmo, avente complessità $O(m + n)$, per questo problema, sottolineando tutti gli aspetti che è necessario considerare per ottenere un tale risultato.

Supponiamo che i nodi del grafo siano numerati da 1 a n . Teniamo un array A che memorizzi, per ogni nodo, il nome della componente connessa nella quale esso si trova, ossia l'indice del minimo nodo cui è connesso. Inizializziamo tutto l'array a zero, per dire che nessun nodo è ancora classificato (se la numerazione dei nodi partisse da zero, ovviamente, per l'inizializzazione si dovrebbe usare un valore del tipo -1 .)

Si scandisce poi l'array, soffermandosi su ogni nodo i per cui $A[i]$ è uguale a zero. Per un tale nodo, si effettua una visita BFS della sua componente connessa; per ogni nodo incontrato n , si pone $A[n] = i$. Alla fine, tutti gli indici dell'array saranno diversi da zero.

A questo punto conviene servirsi di un altro array C nel quale alla fine vogliamo avere $C[i] = 0$ se i non è rappresentante di alcuna componente; $C[i] = k$ se i è rappresentante di una componente di dimensione k . Il calcolo del contenuto di C , facendo uso di A , è banale.

Notando che C contiene numeri compresi tra 0 e n , possiamo ora ordinare i nodi con un algoritmo lineare e stabile, usando come chiave del nodo i proprio $C[i]$. Così facendo, tutti i nodi che non sono rappresentanti finiscono in fondo. Gli altri forniscono l'elenco secondo le specifiche desiderate.

Esercizio 5

Descrivere a parole o con pseudocodice commentato l'operazione di estrazione del minimo nel caso di heap binomiali e di heap di Fibonacci.

Per ciascuno dei due casi, fornire un esempio concreto (mostrando tutte le fasi dell'operazione) di estrazione del minimo da uno heap di 13 elementi, che causi modifiche a tutti gli alberi presenti nella struttura dati.

Ci limitiamo a parlare dell'esempio concreto. Ricordiamo che uno heap binomiale ha "forma" obbligata, ovvero dal numero di elementi possiamo risalire univocamente alle dimensioni dei singoli alberi binomiali che lo compongono. Lo heap in oggetto dovrà dunque avere tre alberi: uno da 8, uno da 4, uno da 1. Se il minimo fosse contenuto nell'albero più piccolo, eliminando quest'ultimo saremmo già posto. Se fosse contenuto in quello da 4, questo si spezzerebbe in $2 + 1$ e potrebbe venir ricomposto, assieme all'albero da 1 che sopravvive, con una fusione dei due alberi da uno, cosicché si avrebbero a quel punto due alberi da 2 a loro volta fusi in un albero da 4. Tutto questo senza toccare l'albero da 8. Nell'esempio il minimo dovrà dunque risiedere nell'albero da 8. Con l'estrazione, esso viene smembrato in $4 + 2 + 1$. Ora i vecchi alberi da 1 e da 4 sono affiancati entrambi da alberi dello stesso tipo; questo basta a garantire che saranno soggetti a fusione, e che quindi alla fine ogni albero sarà stato toccato.

Nel caso degli heaps di Fibonacci, ricordiamo che il layout degli alberi prima dell'estrazione può essere molto più vario. Dopo l'estrazione, comunque, grazie all'operazione di consolidamento, avremo le forme tipiche degli heaps binomiali e quindi, nel nostro caso, ancora una volta un albero da 8 e uno da 4.

Un esempio molto sbrigativo è quello dove partiamo da tredici nodi radice, ad esempio prodotti da tredici inserzioni successive. Se dopo tali inserzioni viene effettuata un'estrazione di minimo, avviene la consolidate che, raggruppando i nodi in $8 + 4$, ovviamente li tocca tutti.