

Primo Scritto

ASD1 2002-2003

Exercise 1 Dire quali delle seguenti affermazioni sono vere. Ove false, fornire un controesempio.

1. se $f(n)$ e $g(n)$ sono definitivamente non negative, allora $f(n) + g(n) = \Theta(\max\{f(n), g(n)\})$.
2. $\Omega(n^2) \cap O(n^2) = \emptyset$.
3. $f(n+1) = O(f(n))$ se $f(n)$ è definitivamente non negativa.
4. $2^{\alpha n} = O(3^n)$ per ogni $\alpha > 0$.

Exercise 2 Dimostrare che $\sum_{t=1}^n \log t = \Theta(n \log n)$.

Exercise 3 Ordinare le seguenti funzioni per ordine di crescita asintotico non decrescente, ove $k > 1$ sia una costante positiva comune. Ve ne sono alcune che presentano lo stesso ordine di crescita? Come cambia la situazione per $k = 1$? $f(n) = n^k$, $f(n) = (100n + 10)^k$, $f(n) = \binom{n}{k}$, $f(n) = k^n$, $f(n) = \log k^n$.

Exercise 4 Un grafo $G = (V, E)$ si dice bipartito se posso bipartizionare V in due sottoinsiemi disgiunti V_1 e V_2 tali che ogni arco in E abbia un estremo in V_1 e l'altro in V_2 .

- Dimostrare che un grafo è bipartito se e solo se non contiene cicli dispari.
- Descrivere un algoritmo con tempo di calcolo $O(m + n)$ (ossia lineare) che dato in input un grafo connesso G restituisca o una dimostrazione che G non è bipartito (ciclo dispari) od una dimostrazione che G è bipartito (una bipartizione valida $V = V_1 \cup V_2$).

Exercise 5 Si assuma che uno Heap binario sia un albero binario i cui nodi ospitino le chiavi e gli indirizzi in memoria degli elementi di cui interessa la gestione, e che goda delle seguenti proprietà:

- 1 è quasi pieno;
- 2 ha la proprietà dell'ordinamento dello heap.

Si supponga per semplicità che tutti gli elementi abbiano chiavi di valore diverso. Spiega in cosa consistano le due proprietà listate qui sopra. Dimostra che uno heap di n elementi ha altezza $\lceil \log_2 n \rceil$. Sai dire in quale nodo debba risiedere l'indirizzo dell'elemento con chiave più piccola e perchè? Supponi di voler rimuovere un elemento a chiave minima mantenendo però le due proprietà dello heap. Descrivere una procedura che esegua tale eliminazione in al più $O(\log_2 n)$ passi. Si supponga di essere nella situazione di avere la proprietà 1 ma non la 2. Descrivere una procedura che porti ad avere entrambe le proprietà in al più $O(n)$ passi. Esprimere l'analisi del tempo di calcolo di tale procedura.