

### Esercizio 1 (\*\*\*)

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.  $4f(n) + 4g(n) \in O(2^{f(n)+g(n)})$

### Esercizio 2 (12\*)

Si indichi con  $Call(n)$  il numero di calls alla function F eseguite dal seguente programma quando riceva il naturale  $n$  come valore numerico in input.

```
#include<iostream.h>
```

```
long long int F(int n) {  
    if(n <= 1) return n;  
    return F(n-1) + F(n-2);  
}
```

```
int main() {  
    int n; cout << "Dammi n: "; cin >> n;  
    cout << "F[" << n << "] = " << F(n) << endl;  
}
```

(\*) Scrivere una ricorrenza e delle condizioni al contorno che definiscano  $Call(n)$ .

(\*\*\*) Individuire il nesso con un'altra ricorrenza famosa, esaminata durante il corso.

(\*\*\*) Avvalersi di tale nesso per caratterizzare l'andamento asintotico di  $Call(n)$ , avvalendosi di quanto già noto per la ricorrenza famosa di riferimento.

(\*\*\*\*\*) Dimostrare la validità dei bounds asintotici utilizzati.

### Esercizio 3 (\*\*\*)

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

$$f_1(n) = 2^n, \quad f_2(n) = n^{\log_2 n}, \quad f_3(n) = n^{\log_3 n}, \quad f_4(n) = (\log_2 n)^n, \quad f_5(n) = (\log_3 n)^n, \quad f_6(n) = (\log_2 n)^{n+1}.$$

### Esercizio 4 (15\*)

Sia dato un grafo non orientato e connesso  $G$  con  $n$  nodi ed  $m$  archi. Si assuma che ad ogni arco  $e$  di  $G$  sia associato un peso  $w_e$ . Si assuma per semplicità che i pesi siano degli interi non negativi. Vogliamo trovare un sottografo connesso di  $G$  per cui sia minima la somma dei pesi degli archi in esso contenuti.

(\*) Si dimostri come sia sempre possibile assumere che tale sottografo sia aciclico, ossia sia un albero.

(\*\*) Descrivere un qualsiasi algoritmo polinomiale per il problema dell'albero di copertura minimo testè introdotto.

(\*\*\*\*\*) Argomentarne la correttezza. (\*) Analizzarne la complessità. (\*\*\*) Dare un algoritmo lineare per lo stesso problema nel caso in cui tutti i pesi siano o degli 1 o dei 2. (\*\*\*) Dimostrarne la correttezza.

### Esercizio 5 (7\*)

(\*) Descrivere a parole o con pseudocodice commentato l'algoritmo di QuickSort.

(\*\*\*\*\*) Sulla base di tale descrizione, esprimere l'analisi del valore atteso del tempo di calcolo.