# Algoritmi e Strutture Dati
## 23 Agosto 2019

Cognome ........................ Nome ........................ Matricola ........

1. Durata: 2:30 h
2. La **leggibilità** è un prerequisito: parti difficili da leggere potranno essere ignorate.
3. Quando si presenta un algoritmo è fondamentale spiegare l'**idea soggiacente e motivarne la correttezza**.
4. L' **efficienza e l'aderenza alla traccia** sono criteri di valutazione delle soluzioni proposte.
5. Si **consegnano tutti i fogli**, con nome, cognome, matricola e l'indicazione **bella copia** o **brutta copia**.

## Domande

**Domanda A** (5 punti) Data la ricorrenza

$$T(n) = \begin{cases} 2 & \text{se } n = 0 \\ T(n-1) + 2n + 1 & \text{se } n > 0 \end{cases}$$

determinarne la soluzione esatta.

**Domanda B** (4 punti) Si consideri una tabella hash di dimensione $m = 8$, e indirizzamento aperto con doppio hash basato sulle funzioni $h_1(k) = k \ mod \ m$ e $h_2(k) = 1 + k \ mod \ (m-2)$. Si descriva in dettaglio come avviene l'inserimento della sequenza di chiavi: 14, 22, 10, 16, 8.

**Domanda C** (5 punti) Si calcoli la lunghezza della longest common subsequence (LCS) tra le stringhe *armo* e *oro*, calcolando tutta la tabella $L[i,j]$ delle lunghezze delle LCS sui prefissi usando l'algoritmo visto in classe.

## Esercizi

**Esercizio 1** (7 punti)

- Scrivere una funzione `Rep(T,k)` che dato un albero di ricerca $T$ e una chiave $k$ ritorna il numero di occorrenze di $k$ in $T$. Valutare la complessità dell'algoritmo definito.
- Se si assume che l'albero binario di ricerca non contenga chiavi ripetute cosa cambia? (Ovviamente in questo caso il numero di occorrenze è al più uno). Adattare la soluzione e discutere la relativa complessità.

**Esercizio 2** (9 punti) Dato un insieme di $n$ numeri reali positivi e distinti $S = \{a_i, a_2, \ldots, a_n\}$, con $0 < a_i < a_j < 1$ per $1 \le i < j \le n$, un *(2,1)-boxing* di $S$ è una partizione $P = \{S_1, S_2, \ldots, S_k\}$ di $S$ in $k$ sottoinsiemi (cioè, $\bigcup_{j=1}^{k} S_j = S$ e $S_r \cap S_t = \emptyset, 1 \le r \neq t \le k$) che soddisfa inoltre i seguenti vincoli:

$$|S_j| \le 2 \qquad \text{e} \qquad \sum_{a \in S_j} a \le 1,\ 1 \le j \le k.$$

In altre parole, ogni sottoinsieme contiene al più due valori la cui somma è al più uno. Dato $S$, si vuole determinare un (2,1)-boxing che minimizza il numero di sottoinsiemi della partizione.

1. Scrivere il codice di un algoritmo greedy che restituisce un (2,1)-boxing ottimo in tempo lineare. (Suggerimento: si creino i sottoinsiemi in modo opportuno basandosi sulla sequenza ordinata.)
2. Si enunci la proprietà di scelta greedy per l'algoritmo sviluppato al punto precedente e la si dimostri, cioè si dimostri che esiste sempre una soluzione ottima che contiene la scelta greedy.

**Nota**: Correzione, risultati e visione dei compiti: 28 Agosto 2019, ore 14:30 (1BC/50)