Appunti lezione – Capitolo 1
Introduzione

Alberto Montresor

21 Agosto, 2014

1 Esercizio: Dimostrare che l'invariante di $\mathsf{min}()$ è rispettata

L'invariante da dimostrare è la seguente: All'inizio di ogni iterazione del ciclo **for**, la variabile *min* contiene il minimo parziale degli elementi $A[1 \dots i-1]$.

- *Inizializzazione*: All'inizio della prima iterazione, il valore della variabile i è pari a 2. Essendo *min* inizializzato ad $S[1]$, esso è banalmente il minimo del sottovettore $S[1 \dots 1]$.
- *Conservazione*: All'inizio di una iterazione, *min* contiene il valore del minimo di $S[1 \dots i-1]$. Al termine, i viene incrementata di 1 e *min* viene confrontanto con $S[i]$, ed eventualmente aggiornato con tale valore, se più basso. Quindi all'inizio del ciclo successivo *min* contiene il minimo di $S[1 \dots i]$, come richiesto.
- *Conclusione*: al termine del ciclo, il valore della variabile i è pari a $n+1$. Quindi *min* contiene il minimo del vettore $S[1 \dots n]$, ovvero il valore che volevamo ottenere.

2 Esercizio: Dimostrare che $\mathsf{binarySearch}()$ è corretta.

La correttezza si dimostra per induzione sulla dimensione n del vettore.

- Passo base: Se $n = 0$, ovvero se $i > j$, il valore cercato non è presente e correttamente l'algoritmo ritorna 0.
- Ipotesi induttiva: vogliamo dimostrare che l'algoritmo è corretto per la dimensione n, e supponiamo di aver dimostrato che l'algoritmo è corretto per tutte le dimensioni $n' < n$.
- Passo induttivo: Sia m l'elemento mediano; se $S[m] = v$, l'elemento è stato trovato e correttamente l'algoritmo ritorna l'indice m. Se invece $S[m] \neq v$, il valore deve necessariamente trovarsi negli elementi $S[i \dots m-1]$ (se minore) o $S[m+1 \dots j]$ (se maggiore). Tali sottovettori hanno dimensione minore di n, e quindi la ricerca dell'elemento in essi dà origine ad un risultato corretto.