

Consigli Tattici

17 maggio 2018

- Usare funzioni inline invece delle macro (che possono creare errori e non danno vantaggi in termini di performance).
- Spezzare in funzioni ogni volta che logicamente ha senso.
- Mettere assert durante la scrittura del codice.
- Seguire le quattro fasi: understand-plan-do-check.
- Chiedersi: qual'è il problema più simile a questo che ho già visto?
- Partire cercando soluzioni ai subtask: spesso i subtask sono messi per facilitare la risoluzione graduale del problema.
- Pianificare un tempo massimo da dedicare a un problema.
- Se c'è un passaggio complicato o in qualche senso contoso (una dinamica con un po' di off by one possibili, tanti casi da gestire...) scrivi prima tutto su un foglio di carta e poi ricopia. Quando lo fai su carta ti sembrerà di sprecare tempo, ma poi ricopiare è molto più veloce di quello che credi e ammazzerai i tempi necessari per debuggare.
- Dedica qualche minuto in più a pulire, semplificare e capire in profondità una soluzione funzionante, prima di iniziarne l'implementazione. A volte qualche minuto in più di pensiero risparmia mezz'ora di debugging.

