

Alcune considerazioni sul progetto di soluzioni software



Fondamenti di Informatica L- A

Criteri da tenere a mente

- Lo **stile di programmazione** è importante
 - leggibilità (commenti / nomi significativi di variabili / indentazioni /...)
 - modularità e riuso (funzioni / prototipi / librerie standard / ...)
 - adozione di convenzioni (es. nomi di funzioni con la minuscola, ...; proprie di ciascun linguaggio)
- La fase di analisi/progetto **precede** la fase di implementazione...
 - in generale: metodologie (Ingegneria del Software)
 - in questo corso: alcune considerazioni di base per problemi su piccola scala

Fondamenti di Informatica L- A

Basi di analisi di algoritmi

- Algoritmi equivalenti: alcuni più “efficienti” di altri
- Cosa vuol dire “efficiente”?
 - Tempo di esecuzione / spazio di memoria utilizzato?
 - In relazione alla dimensione dei dati di input?
 - Caso medio / caso peggiore?
 - In funzione di una o più **operazioni elementari**
 - Indipendentemente dalla velocità della macchina (complessità computazionale)
 - Dipendentemente dai vincoli imposti dall’architettura (scelte di progetto)
 - Ordini di grandezza: notazione O-grande

Fondamenti di Informatica L- A

Ricerca lineare

```
int find(int A[], int n, int key) {
    int i;
    for(i=0; i<n; i++)
        if (A[i]==key)
            return i;
    return -1;
}
```

- Caso peggiore: **Complessità dell'algoritmo: $O(n)$**
 - **key** non è presente in **A** (o è l'ultimo elemento)
 - in questo caso: **n** iterazioni (operazioni di confronto)
- Caso medio:
 - **key** è a metà di **A**
 - in questo caso: **n/2** operazioni di confronto

Fondamenti di Informatica L- A

Ricerca binaria (su array ordinato)

```

int find(int A[], int n, int key) {
    int mid;
    if( n < 0 ) return -1; // not found
    mid = n/2;
    if( key > A[mid] )
        return find( A+mid+1, n-mid-1, key );
    else if( key < A[mid] )
        return find( A, mid-1, key );
    else return mid;
}

```

Complessità dell'algoritmo: $O(\log n)$

- Caso peggiore:
 - **key** non è presente in **A**
 - a ogni iterazione l'intervallo si dimezza $\rightarrow \log_2 n$ iterazioni per arrivare a \emptyset
 - in questo caso: **log n iterazioni**

Complessità: perché è importante...

- Esempio visto: ricerca di un elemento in un vettore di n elementi. Caso medio:
 - Ricerca binaria: $\log n - 1$
 - Ricerca lineare: $n/2$
 - Oracolo: 1

n	Oracolo	Ric. binaria	Ric. lineare
10	1	3	4
10^2	1	6	50
10^6	1	19	500mila
10^{10}	1	39	5 miliardi

Quando bisogna tenere conto dell'architettura

- Esistono vari tipi eterogenei di operazioni elementari, il cui costo dipende significativamente dall'architettura
- Caso tipico: elaborazione di dati su file
- Configurazione tipo:
 - File system su HD con latency time ~ 10 ms
 - Transfer rate HD: 10MBps (100 ms/MB)
 - Transfer rate RAM (133MHz, 64bit): 1GBps (1 ms/MB)
 - Memoria disponibile >> dimensione dati
- Punti chiave di una generica soluzione:
 1. Lettura da file
 2. Trasformazione dati
 3. Scrittura risultato su file

Fondamenti di Informatica L- A

Scelte di progetto ed efficienza

- Costo del trasferimento file \leftrightarrow memoria:
 - Latency + size / transfer rate ($\sim 10\text{ms} + 100 \text{ ms/MB}$)
- Costo del trasferimento dati memoria \leftrightarrow processore:
 - Latency + size / transfer rate ($\sim 20\text{ns} + 1 \text{ ms/MB}$)
- Costo per effettuare una operazione semplice:
 - Frequenza CPU ($\sim 1 \text{ GHz} \rightarrow 1 \text{ ns}$)

Variabili

F = dimensione del file (MB)

D = dimensione del blocco di dati da trasferire (MB)

n = numero di record

M = numero di iterazioni

Fondamenti di Informatica L- A

Scelte di progetto ed efficienza

- Costo del trasferimento file \leftrightarrow memoria (ms):
 - $F \sim 10^{-1} \rightarrow \sim 10$
 - $F \gg 10^{-1} \rightarrow \sim 100 \times F$
 - Costo del trasferimento dati memoria \leftrightarrow processore (ms):
 - $D \sim 10^{-5} \rightarrow \sim 10^{-5}$
 - $D \gg 10^{-5} \rightarrow \sim D$
 - Costo per effettuare una operazione semplice:
 - $\sim 10^{-9}$
- 2-6 ordini di grandezza
4-9 ordini di grandezza
- Variabili**

F = dimensione del file (MB)
D = dimensione del blocco di dati da trasferire (MB)
n = numero di record
M = numero di iterazioni

Fondamenti di Informatica L- A

1 Problema, 2 Scenari, 3 Soluzioni

- Programma che gestisce un DB i clienti di un negozio
 - il DB contiene per ogni cliente: nome, cognome, ..., spesa2006
 - a ogni acquisto, il DB deve aggiornare la spesa2006 del cliente
 - il programma deve conservare i dati in modo permanente
 - Scenario 1: $F = 100\text{MB}$, $n = 10^6$
 - Scenario 2: $F = 100\text{KB}$, $n = 10^3$
- [Soluzione A]
- A.1: leggi tutto il file su una tabella in memoria
 - A.2: esegui tutte le operazioni (M operazioni)
 - A.3: alla fine, salva la tabella aggiornata su file
- [Soluzione B] per ciascuna delle M operazioni:
- B.1: cerca sul file il record interessato
 - B.2: esegui l'operazione sul record
 - B.3: salva il singolo record su file
- [Soluzione C] \rightarrow varianti di [B]:
- [C.I] il file è ordinato \rightarrow è possibile effettuare una ricerca binaria
 - [C.II] esiste un oracolo che, dato un cliente, restituisce la posizione del relativo record all'interno del file

Fondamenti di Informatica L- A

Considerazioni conclusive

- **Impatto della complessità degli algoritmi scelti sulla soluzione**
 - importante sapere costo di algoritmi di uso comune!
es: ricerca (binaria), ordinamento, ...
- **Impatto delle scelte progettuali**
 - utilizzo di tabelle può essere necessario
 - approcci misti: suddividere file in porzioni (fare per esercizio...)
- **Attenzione: scenario molto semplificato!**
 - presenza/dimensione di eventuali cache
 - architetture con parallelismi
 - limiti sulla memoria disponibile

Fondamenti di Informatica L- A

Preparazione all'esame



Fondamenti di Informatica L- A

Prova pratica

- Facile prendere due punti se
 - avete seguito il laboratorio
 - riuscite a risolvere gli esercizi proposti
- Punti chiave:
 - avere dimestichezza con l'utilizzo dell'ambiente di programmazione del laboratorio (editor, compiler)
 - saper utilizzare il debugger
 - aver fatto sufficiente pratica, anche su problemi semplici (soprattutto: vettori, cicli, funzioni)

Fondamenti di Informatica L- A

Prova scritta

- Per raggiungere la sufficienza occorre raggiungere un punteggio minimo in ciascuna domanda, come da tabella:

Domanda	Punti	Soglia
1. Progetto	18	9
2. Analisi	7	3
3.4. Teoria	4+4	4

Fondamenti di Informatica L- A

Valutazione di un progetto

- Presenza di un progetto
- Giustificazione delle scelte progettuali
- Correttezza della soluzione
- Leggibilità del codice (commenti, indentazione)
- Modularità (uso di funzioni) → non è sempre necessario/richiesto sviluppare la soluzione per intero!
- Qualità degli algoritmi
- Capacità di usare gli strumenti in modo appropriato
 - visibilità e tempo di vita delle variabili
 - passaggio dei parametri
 - corretto utilizzo delle istruzioni su file/stringa
 - implementazione dei cicli
 - ...

Fondamenti di Informatica L- A

Come esercitarsi per lo scritto (domanda di progetto)

- Sul sito web: testi e soluzioni di esami dell'anno scorso. Domande di progetto tipo:
 - dato un problema (di media difficoltà), progettare una possibile soluzione e implementare una parte delle funzioni, o una specifica funzione
 - dato un problema (facile), progettare una possibile soluzione e implementarla in modo completo

Fondamenti di Informatica L- A

Come esercitarsi per lo scritto (domanda di progetto)

- Consigli:
 1. Non leggere direttamente la soluzione!
 2. Esercitarsi a
 - Disegnare lo schema di un progetto coerente di soluzione, in cui sia chiaro – ad esempio – il ruolo delle variabili globali e/o delle eventuali variabili dinamiche. Esprimere considerazioni di progetto. Eventualmente, mettere a confronto varie possibili scelte.
 - Sviluppare codice modulare, distribuito in più funzioni (anche se non sono tutte implementate).
 - Scrivere i prototipi e ragionare su un “progetto” globale. Bisogna essere in grado di ragionare su funzioni (anche non implementate), in termini di:
 - parametri di input
 - valore di uscita
 - gestione dei casi di errore
 - utilizzo di strutture dati esterne (es. variabili globali)
 3. Implementare il codice scritto e usare il debugger per ispezionarlo

Fondamenti di Informatica L- A

Come esercitarsi per lo scritto (domanda di analisi)

- Sul sito web: testi e soluzioni di esami dell'anno scorso. Domanda di analisi tipo:
 - data una funzione ricorsiva, indicare come si comporta con un dato input. **Giustificare accuratamente la risposta.**
- Consigli:
 1. Esercitarsi a giustificare accuratamente la risposta. Vedi sul sito web.
 2. Implementare le funzioni proposte gli anni passati (e/o altre di propria iniziativa) e usare il debugger per ispezionarne il funzionamento
- Errore tipico: presente solo l'output della funzione, senza giustificazione

Fondamenti di Informatica L- A

Come esercitarsi per lo scritto (domanda di teoria)

- Sul sito web: testi e soluzioni di esami dell'anno scorso. Domanda di teoria tipo:
 - Si parli in modo esauriente e sintetico dell'argomento X.
- Consigli:
 1. Studiare non (solo) sui lucidi (libri, biblioteche, internet, ...)
 2. Leggere bene la domanda...
 3. "Esauriente e sintetico" significa:
 - mostrare padronanza dell'argomento
 - toccare gli aspetti chiave
 - non andare fuori tema
- Errore tipico:
 - si presentano solo gli aspetti che si ricordano, ma non i più importanti
 - manca una visione complessiva dell'argomento
 - si dedica spazio alla trattazione di argomenti non richiesti