Introduzione agli algoritmi

Un algoritmo è una sequenza di comandi elementari ed univoci che terminano in un *tempo finito* ed operano su *strutture dati* (quest’ultime memorizzano i dati semplificandone l’accesso e la modifica).

Ci sono diverse strutture dati ed hanno diverse applicazioni, ognuna di esse è utilizzata per risolvere specifici problemi meglio di altre.

Un algoritmo deve essere *efficiente*, è quindi necessario che finisca la computazione in un tempo “ragionevole”, l’obbiettivo è risolvere un problema nel minor tempo possibile, quindi ci occuperemo del cosiddetto *costo computazionale* in termini di tempo e spazio.

Esempio :

Dobbiamo ordinare n = numeri interi. Abbiamo due calcolatori :

* Il calcolatore V (veloce) effettua operazioni al secondo.
* Il calcolatore L (lento) effettua operazioni al secondo.

Abbiamo poi due diversi algoritmi di ordinamento :

* L’algoritmo IS (Insertion sort) richiede operazioni.
* L’algoritmo MS (Merge sort) richiede operazioni.

Se dovessimo utilizzare l’algoritmo più lento sul calcolatore più rapido, e l’algoritmo più veloce sul calcolatore più lento, la maggiore efficienza di V riuscirebbe a bilanciare la minor efficienza dell’algoritmo IS? Vediamo con il calcolo dei tempi :

È chiaro che un algoritmo più veloce performi meglio anche su un computer lento, rispetto ad un algoritmo lento su un computer più veloce.

Adesso aumentiamo la quantità di dati da computare a

Un algoritmo più lento, all’aumentare dei dati da computare, aumenta esponenzialmente il suo tempo di esecuzione.

Problem Solving

“Passare da un problema alla soluzione”   
Approccio al problem solving :

1. Analisi del problema, comprensione e identificazione del problema
2. Esplorazione degli approcci possibili tramite i metodi noti
3. Selezionare un approccio
4. Definizione dell’algoritmo risolutivo, identificando i dati
5. Riflessione critica, ripensando alla soluzione proposta.

Un algoritmo si può definire *corretto* se per ogni istanza di un problema computazionale, termina con l’output corretto.

Inoltre per valutare l’efficienza di un algoritmo, dobbiamo analizzarlo senza che l’analisi sia influenzata da una specifica tecnologia. Usiamo quindi una macchina teorica, **la Random access machine**, ha un singolo processore e svolge operazioni elementari, ciascuna di esse richiede per definizione un tempo costante. Esiste un limite alla dimensione di ogni valore memorizzato ed al numero complessivo di valori utilizzati.

Criterio della misura di costo uniforme

Se ogni dato in input sia un valore minore di k = , ciascuna operazione elementare sui dati del problema verrà eseguita in un tempo costante, in tal caso si parla di misura di costo uniforme. In un contesto realistico però, è comune che un dato sia più grande del valore k, si usano quindi più parole.