16 Teoria della complessità

Come abbiamo visto nella definizione 15.1 dividiamo i problemi in diverse categorie.
I problemi che vedremo in questa sezione sono **presumibilmente intrattabili**, ovvero che abbiamo a disposizione solo algoritmi di costo *esponenziale* per risolverli ma che nessuno ha dimostrato effettivamente che non possano esistere algoritmi polinomiali.

Esempio 16.1 (Problema della clique)**.** Dato un grafo $F = (V, E)$ e un intero $k > 0$, stabilire se G contiene un clique (sottografo completo) di almeno k nodi.

Esempio 16.2 (Problema del cammino (o ciclo) Hamiltoniano)**.** Dato un grafo $G = (V, E)$, trovare un cammino (o ciclo) semplice che passa da tutti i vertici di G una ed una sola volta.

16.1 Velocità dei calcolatori

Studiamo ora la dimensione dei dati trattabili in funzione dell'incremento della velocità dei calcolatori per dimostrare che lo sviluppo tecnologico non riesce a bilanciare un algoritmo inefficiente.

Dati due calcolatori C_1, C_2 con C_2 k volte più veloce di C_1. Il tempo di calcolo a disposizione è t e:

- n_1 rappresenta i dati trattabili nel tempo t su C_1
- n_2 rappresenta i dati trattabili nel tempo t su C_2

Osservazione 16.1.1. Usare C_2 per un tempo t equivale a usare C_1 per un tempo $k \cdot t$.

Algoritmo polinomiale che risolve il problema in $c \cdot n^s$ secondi (c, s costanti).

- C_1: $c \cdot n_1^s = t \longrightarrow n_1 = (\frac{t}{c})^{\frac{1}{s}}$
- C_2: $c \cdot n_2^s = t \longrightarrow n_1 = (k \cdot \frac{t}{c})^{\frac{1}{s}} = k^{\frac{1}{s}} \cdot (\frac{t}{c})^{\frac{1}{s}}$

Concludiamo quindi che il miglioramento è di un fattore moltiplicativo $K^{\frac{1}{s}}$. Ad esempio per $k = 10^9$ e $s = 3$ i dati trattabili saranno moltiplicati per 10^3.

16.2 Tipi di problemi

I tipi di problemi che possiamo studiare sono i seguenti:

- **Problemi decisionali**: richiedono una risposta binaria, ad esempio determinare se un numero è primo
- **Problemi di ricerca**: data un'istanza x, richiedono di restituire una soluzione s, ad esempio trovare un cammino tra due vertici.
- **Problemi di ottimizzazione**: data un'istanza x, si vuole trovare la *migliore* soluzione s tra tutte le soluzioni possibili. Ad esempio la ricerca della clique di dimensione massima.

16.3 Problemi decisionali

Nella teoria della complessità si studiano solamente i problemi **decisionali**, questo perché:

- Essendo la risposta binaria, non ci si deve preoccupare del tempo richiesto per restituire la soluzione e quindi tutto il tempo è speso per il calcolo
- La difficoltà di un problema è già presente nella usa versione decisionale. Tutti i problemi di ottimizzazione sono esprimibili in forma decisionale, chiedendo l'esistenza di una soluzione che soddisfi una certa proprietà. Il problema di **ottimizzazione** è quindi almeno tanto difficile quanto quello decisionale e quindi mi basta caratterizzare la complessità di quest'ultimo per dare un limite inferiore alla complessità del primo.

16.4 Classi di complessità

Dato un problema decisionale Π ed un algoritmo A, diciamo che A risolve Π se, data un'istanza di input x

$$A(x) = 1 \iff \Pi(x) = 1$$

A risolve P in tempo $t(n)$ e spazio $s(n)$ se il tempo di esecuzione e l'occupazione di memoria di A sono rispettivamente $t(n)$ e $s(n)$. Data una qualunque funzione $f(n)$:

- $Time(f(n))$: insieme dei problemi decisionali che possono essere risolti in **tempo** $O(f(n))$.
- $Space(f(n))$: insieme dei problemi decisionali che possono essere risolti in spazio $O(f(n))$

16.4.1 Classe P

Definizione 16.1 (Algoritmo polinomiale in tempo)**.** *esistono due costanti c, n0 ¿ 0 t.c. il numero di passi elementari è al più nc per ogni input di dimensione n e per ogni n ¿ n0*

Definizione 16.2 (Classe P)**.** *è la classe dei problemi risolvibili in tempo polinomiale nella dimensione n dell'istanza di ingresso*