

L31 - 06/12/2024

Complessità computazionale

Dato un problema, si cerca di costruire l'algoritmo risolutivo. Si fa poi una stima delle risorse (generalmente tempo e spazio) utilizzate dall'algoritmo.

È però possibile fare, partendo direttamente dal problema, una stima delle risorse necessarie e sufficienti per la risoluzione del problema stesso.

- Si vuole arrivare al miglior algoritmo possibile

Per risorse necessarie (limitazione inferiore) e sufficienti (limitazione superiore) si intende quanta risorsa r sia necessaria e sufficiente per risolvere Π , dove:

- r risorsa computazionale (tempo, spazio, ecc)
- Π problema risolubile algebricamente

Limitazione superiore

$f : \mathbb{N} \rightarrow \mathbb{N}$ funzione

$f(n)$ risorsa r è **sufficiente** per risolvere Π se esiste un algoritmo \mathcal{A} che risolve Π utilizzando su ogni input di lunghezza n al più $f(n)$ risorsa r ($\forall n \geq 0$)

- dominio lunghezza input
- codominio risorse

Limitazione inferiore

$f : \mathbb{N} \rightarrow \mathbb{N}$ funzione

$g(n)$ risorsa r è necessaria per risolvere Π se per ogni algoritmo \mathcal{A} che risolve Π esiste un input di lunghezza n ($\forall n \geq 0$) su cui \mathcal{A} utilizza almeno $g(n)$ risorsa r

Esempio

$\Pi = \text{sort}$, $r = \# \text{confronti}$

Algoritmo Bubblesort: $\Theta(n^2)$ confronti

Si è dimostrato che ogni algoritmo di ordinamento basato su confronti utilizza $\Omega(n \log n)$ nel caso peggiore (limitazione inferiore)

[...]

Complessità computazionale

Classificazione dei problemi in base alle risorse utilizzate per la loro soluzione

Classe di complessità

Insieme di problemi che possono essere risolti utilizzando la stessa quantità di una determinata risorsa

Classe P

Classe dei problemi risolvibili in tempo polinomiale rispetto alla lunghezza dell'input (*definizione approssimativa*)

Alcuni problemi appartenenti a questa classe sono il prodotto di matrici, l'ordinamento di un vettore, l'albero ricoprente minimo

Il problema dello zaino, la colorazione di grafi e il commesso viaggiatore non rientrano in questa classe di problemi

- L'algoritmo greedy per risolvere il problema dello zaino non sempre dà la soluzione corretta

Problema $\Pi \subseteq I \times S$ dove

- I è l'universo delle possibili istanze (input)
- S è l'universo delle soluzioni

$(x, s) \in \Pi$ se e solo se s è una soluzione di Π su input x

Albero ricoprente

I = grafi non orientati

S = alberi

$(x, s) \in \Pi$ se e solo se s è un albero ricoprente per il grafo x (*problema di ricerca*)

Albero ricoprente minimo

I = grafi non orientati pesati

S = alberi

$(x, s) \in \Pi$ se e solo se s è un albero ricoprente per il grafo x minimo: tra tutti gli alberi s tali che $(x, s) \in \Pi$ cerchiamo s^* tale che $\text{peso}(s^*) \leq \text{peso}(s)$ (*problema di ottimizzazione*)

Tipologie di problemi

Ricerca

Data $x \in I$ trovare $s \in S$ tale che $(x, s) \in \Pi$

Ottimizzazione

Data $x \in I$ trovare $s \in S$ tale che $(x, s) \in \Pi$ che soddisfi un criterio di ottimalità fissato (*es min/max*)

Decisione

Si ottiene una risposta binaria, con quindi $S = \{0, 1\}$

[...]

I problemi di decisione sono una versione più semplice che però ci fornisce informazioni sul problema generale. Vantaggi dei problemi di decisione:

- Risolvendo il problema di decisione si ha in output un formato binario, che è molto semplice
- Il problema di ottimizzazione è difficile almeno quanto il problema di decisione
- Considerando solo questi problemi con output binario, è possibile confrontare tra di loro problemi che si occupano di ambiti diversi

L31 - 09/12/2024

[...]

L32 - 11/12/2024

[...]