Capitolo 1.4 - 1.5

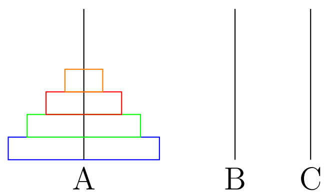
Lorenzo Vaccarecci

4 Marzo 2024

1 Correttezza di algoritmi ricorsivi

1.1 Torre di Hanoi

Devo spostare tutti i dischi da A a C, senza mai mettere un disco più grande su uno più piccolo.



Pensiamo in modo ricorsivo:

- $n = 1 \rightarrow$ da A a C
- $n + 1 \rightarrow$ sposto $1 \ldots n$ da A a B, sposto $n + 1$ da A a C, sposto $1 \ldots n$ da B a C

```
Hanoi(n, from, aux, to) {
    if(n==1) move(from, to);
    else {
        Hanoi(n-1, from, to, aux); // T(n-1)
        move(from, to); // 1
        Hanoi(n-1, aux, from, to); // T(n-1)
    }
}
```

Complessità: $T(n)$= numero di mosse(move) per n dischi. $\rightarrow$ Lo possiamo esprimere ricorsivamente (induttivamente):

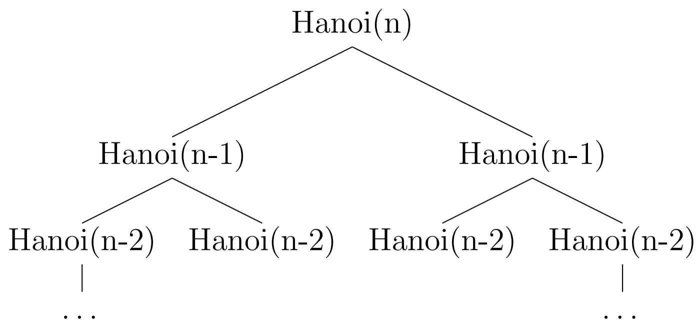
Relazione di ricorrenza

$T(1) = 1$

$T(n > 1) = T(n - 1) + 1 + T(n - 1)$

Una tecnica che spesso consente di individuare la soluzione di una relazione di ricorrenza consiste nell'espandere successivamente le chiamate ricorsive nella definizione di $T(n)$, chiamata anche *metodo "empirico"*.

Un modo alternativo di ottenere lo stesso risultato è espandendo ”per livelli” (dell'albero di ricorsione), nel modo illustrato sotto:



Una volta individuata la soluzione in uno dei due modi, si può verificare rigorosamente la correttezza per induzione aritmetica.
Si ha quindi che $T(n) = \Theta(2^n)$, mentre la complessità in spazio corrisponde all'altezza dell'albero di ricorsione, ossia alla massima profondità dello stack, quindi è $S(n) = \Omega(n)$.
Induzione aritmetica:

- n=1: 2-1=1
- n>1:
  - Ipotesi induttiva: mosse richieste per n dischi $2^n - 1$
  - mosse per $n + 1$ dischi: $2^{n+1} - 1$

Complessità del problema $O(2^n)$ e $\Omega(2^n)$ → algoritmo esponenziale quindi problema intrattabile (e chiuso). Esiste un algoritmo migliore? No.
Prova: per induzione aritmetica, per spostare n dischi ci vogliono almeno $2^n - 1$ mosse. Vero per $n = 1$
Ipotesi induttiva: per spostare n dischi ci vogliono $2^n - 1$ mosse.
Tesi: per spostare $n + 1$ dischi ci vogliono $2^{n+1} - 1$ mosse.