Lezione 27 Settembre

Lo studio si focalizza sul caso medio degli algoritmi.

Lucido 1

Le formule presentate sono delle relazioni di ricorrenza. Esprimono un valore in funzione di altri valori precedenti.
Formula1 : Numero medio di confronti del quicksort.
Formula2 : Numero medio di scambi del quicksort con le stesse caratteristiche delle algoritmo.
Formula3 : Numero medio di confronti con il perno scelto come mediano tra tre.
Formula4 : Variante del quicksort con l'insertionSort con file di piccole dimensioni. Con l'analisi si può determinare il
valore di n in modo tale che l'algoritmo sia più veloce.

Lucido 2

Programma del corso.
Funzioni generatrici sono uno strumento per trasformare le relazioni di ricorrenza in funzioni risolvibili.
Il giovedì si andrà in laboratorio per approfondire i concetti teorici e verificarli con Maple. Si potrà simulare
gli algoritmi su file di grandi dimensioni.
Metodo simbolico è applicato nell enumerazioni di linguaggi.

Lucido 3

Formula1 : Funzione generatrice del numero medio di confronti nel quicksort.
Rappresentazione finita del problema.
Andando a sviluppare la funzione in serie su una costante, che rappresenza la lunghezza della lista, si trova
il valore medio di confronti utilizzando il quicksort.
H_n+1 è il numero armonico che può essere associato al log di n se n è grande.
Formula2 : La scelta del perno non cambia la complessità, ma la costante

moltiplicativa da 2 passa a 12/7

Lucido 4

Password PAA1617
Il materiale viene passato dai professori, il secondo libro si può usare per consultazione.
Esame: progetto in maple o altri sistemi (libreria python).

Lucido 5

Definizioni

Lucido 6

La complessità basata su confronti è theta nlog. Si cercherà di si far vedere che non è possibile fare di meglio.

Lucido 7

Se si dimostra che il mergesort ha una complessità di nlogn, si può dire che non la complessità dell'ordinamento non
può essere minore di nlogn.

Lucido 8

Per studiare la complessità si usa Cn cioè i confronti. Si cerca una ricorrenza per la funzione Cn. Ci si basa sull'
applicazione effettiva dell' algoritmo, ci si basa cioè sulla divisione a metà del problema.
$C_n = C_(n/2) + c(n/2) + n$
In cui n indica i confronti che faccio durante la fase di fusione.
C_1 è la condizione iniziale.
Dimostrazione: si ipotizza che n è una potenza di 2. Una volta trasformato si può dividere per 2^m.
Si può iterare il procedimento analogo a $X_m = X_{m-1} + 1$ che si può

trasformare in $X_{m-2} + 2$.
Si arriva a $C_{2^0}/2^0 + m$ cioè m.
In un caso particolare si è dimostrato che la complessità è $nlog(n)$. Ma si può dimostrare che anche nel caso medio ha la
stessa complessità.

Lucido 9

Non si può fare di meglio. Per dimostrarlo si usa l'albero binario. Ai nodi esterni sono associate tutte le permutazioni
(cioè n! possibilità). Altezza dell albero mi dice qual è il costo per da ordinare un vettore di lunghezza n.
Si deve quindi sviluppare l'espressione $n! < 2^h$.
Il fattoriale di n grande si può approssimare con la formula di Stirling.
Nel lucido altezza dell albero dei confronti si sostituisce n! con la formula di Stirling. Si cambia di base il log a base 2
per aiutare i calcoli. Si usa la proprietà del prodotto dei log. log di 2π va via perché è un numero finito mentre ln(1/e) = -1
e va a moltiplicare n quindi diventa negativo. Una volta svolto si fa solo di nuovo il cambio di base a 2.