

1. (8/10/12). Introduzione al corso.
2. (10/10/12). Il problema del calcolo dell' n -esimo numero di Fibonacci. Un algoritmo numerico e un algoritmo ricorsivo. Analisi della complessità temporale dell'algoritmo ricorsivo. Un algoritmo iterativo di complessità temporale $O(n)$ e di complessità spaziale $O(n)$ (Fibonacci3). Portare la memoria a $O(1)$: Fibonacci4. Introduzione informale alla notazione asintotica. Algoritmo con complessità $O(\log n)$ per il calcolo dell' n -esimo numero di Fibonacci.
3. (15/10/12) Discussione della complessità spaziale degli algoritmi ricorsivi Fibonacci2 e Fibonacci6. Modello di calcolo RAM. Costi uniformi e logaritmici. Notazioni asintotiche: O -grande, Ω -grande, Θ . O -piccolo, Ω -piccolo. Definizioni e semplici esempi. Proprietà. Esercizi notazioni asintotiche: dimostrazione della transitività della notazione $O()$, e discussione sul fatto che relazione $O()$ non si preserva passando all'esponente (Es. 1).
4. (17/10/12) Complessità caso peggiore, migliore, medio. Il problema della ricerca di un elemento in un insieme: ricerca sequenziale e ricerca binaria. Delimitazioni superiori e inferiori di algoritmi e problemi. Ricerca binaria: versione ricorsiva. Equazioni di ricorrenza. Metodo dell'iterazione. Metodo della sostituzione. Presentazione dell'ultimo problema del Problem Set 1: spezzare efficientemente una tavoletta di cioccolata.
5. (22/10/12) Teorema Fondamentale delle Ricorrenze (Master). Semplici esempi. Quando non si può applicare. Metodo del cambiamento di variabile. Metodo che usa l'albero della ricorsione.
6. (24/10/12) Equazioni di ricorrenza: scenari meno comuni. Embedding di albero binario completo su una griglia bidimensionale (Es. 2). Picture-Hanging Puzzles, ovvero come appendere un quadro in modo perverso arrotolando un corda intorno a dei chiodi in modo tale che, rimuovendo uno qualsiasi dei chiodi, il quadro cada. Soluzione per due chiodi. Soluzione ricorsiva per n chiodi che usa corda esponenzialmente lunga. E soluzione che usa corda di lunghezza polinomiale (Es. 3).
7. (29/10/12). Problema dell'ordinamento. Selection Sort. Insertion Sort: due varianti. Algoritmo di ordinamento MergeSort.
8. (31/10/12). Esercitazione. Discussione sulla velocità asintotica della somma e del prodotto di due funzioni. Esercizio: mettere in ordine le funzioni per velocità crescenti. Esercizio di progettazione di un algoritmo che, dato un vettore ordinato A di n interi distinti e un valore x , trovare (se esistono) due elementi di A che sommano a x . Soluzione banale con complessità quadratica, soluzione di complessità $O(n \log n)$ e soluzione con tempo $O(n)$ (Es. 4).

9. (05/11/12). Algoritmo di ordinamento QuickSort: analisi del caso peggiore, migliore, e intuizioni sul caso medio. Discussione versione randomizzata del QuickSort. Delimitazione inferiore al numero di confronti per il problema dell'ordinamento.
10. (07/11/12). Algoritmo di ordinamento HeapSort.
11. (12/11/12). Esercitazione. Correzione di degli esercizi 2 e 3 del Problem Set 2.
12. (14/11/12). IntegerSort e BucketSort. Discussione semplice soluzione con complessità $O(nk^2)$ per costruzione oracolo (per problema delle slide). RadixSort.
13. (19/11/12). Strutture dati elementari: rappresentazioni indicizzate e rappresentazioni collegate. Implementazione di un dizionario con array ordinato/non ordinato e lista ordinata/non ordinata. Rappresentazioni di alberi. Algoritmi di visita di un albero: profondità versione iterativa, profondità versione ricorsiva (preordine, postordine, ordine simmetrico), ampiezza. Algoritmo per calcolare l'altezza di un albero.
14. (21/11/12). Esercitazione sulle visite di alberi. Ricostruire un albero, dati gli ordini di visita simmetrica e in preordine dei nodi (Problema 3.7 del libro di testo). Dimostrazione che la sequenze di visita in preordine più quella in postordine non sono sufficienti in generale per ricostruire l'albero. Progettazione di un algoritmo che, preso un albero con chiavi e colori (rosso e nero), trova il valore del cammino rosso di tipo nodo-radice di valore massimo (Es 5).
15. (26/11/12). Alberi binari di ricerca. Definizione. Visita in ordine simmetrico di un BST. Ricerca, inserimento, cancellazione (ricerca del massimo, del minimo, del predecessore e del successore di un nodo).
16. (28/11/12). Alberi AVL: definizione ed esempi. Alberi di Fibonacci. Dimostrazione della delimitazione superiore dell'altezza di un albero AVL. Operazioni sugli alberi AVL: search, insert, delete.
17. (03/12/12). Esercitazione. Dato un array di n valori reali. Trovare la coppia di indici i e j con $i < j$ che massimizza $A[j]-A[i]$ (Es. 6). Esercizio su albero AVL e "distanza" fra chiavi (Es. 7).
18. (05/12/12). Correzione Problem Set 3.
19. (10/12/12). Richiami sulla tecnica *divide et impera*. Tecnica della programmazione dinamica. Calcolo della distanza fra due stringhe di caratteri. Descrizione del problema/esercizio: Aiutate il Re Imprenditore (Es. 8).
20. (12/12/12). Code con priorità. Soluzioni semplici con array e liste. Heap e d-heap.

21. (17/12/12). Operazione di merge su d-heap, due implementazioni: generalizzazione dell'algoritmo heapify e algoritmo inserimenti ripetuti. Heap binomiali. Cenni sugli heap di Fibonacci.
22. (19/12/12). Esercitazione sulla programmazione dinamica. Correzione dell'esercizio Aiutate il Re Imprenditore (Es. 8) (con algoritmo lineare che, dato il vettore dei valori degli ottimi dei sottoproblemi, ricostruisce la soluzione ottima). Definizione del problema dell'insieme indipendente di valore massimo su grafi. Algoritmo (polinomiale) esatto per cammini. Algoritmo (polinomiale) esatto per alberi (Es. 9 e 10).
23. (07/01/13). Esercitazione. Il problema del distributore automatico: minimizzare il numero di monete nel dare il resto (Es. 11). Algoritmo greedy (goloso): non sempre funziona. Algoritmo di programmazione dinamica che risolve il problema. Esercizio su visite di alberi (Es. 12): dato albero binario con chiavi nei nodi, progettare algoritmo che dice se esiste un cammino radice foglia con chiavi strettamente crescenti.
24. (09/01/13). Correzione Problem Set 4.