

ASD – Strutture di dati

1. Strutture di dati

Strutture dati astratte

Definizioni

Sequenze

Insiemi

Dizionari

Alberi e grafi

Strutture dati elementari

Liste

Pile

Code

2. Alberi

Introduzione

Definizioni

Alberi binari

Introduzione

Alberi strutturalmente diversi

3. Alberi binari di ricerca

Operazioni ABR

Successore / Predecessore

ABR bilanciati

Definizioni

Alberi Red-Black

4. Grafi

Introduzione

Definizioni

Visite dei grafi

DFS - DAG

Classificazione degli archi

Ordinamento topologico

Componenti fortemente connesse

5. Hashing

Introduzione

Definizioni

Tabelle ad accesso diretto

Funzioni hash

Funzioni hash perfette

Uniformità semplice

Gestione delle collisioni

Indirizzamento aperto

7. Code con priorità

Definizione

Vettore heap

Heap

Alberi binari

Alberi binari heap

8. Insiemi disgiunti

Introduzione

Definizione

Rappresentante

Euristiche

Alberi: euristica sul rango

Contenuti

[1.][pdf-1] Strutture di dati

Strutture dati astratte

Definizioni

Dati

- Dato ¹
- Tipo di dato primitivo ²

- Tipo di dato astratto ³
- Specifica di un TDA ⁴
- Implementazione di un TDA ⁵

Strutture

- Struttura di dati ⁶

Sequenze

- Sequenza ⁷

Insiemi

- Insieme ⁸

Dizionari

- Dizionario ⁹

Alberi e grafi

- Albero ordinato ¹⁰
- Grafo ¹¹

Strutture dati elementari

Liste

- Lista concatenata 12
- Sequenza di nodi 13

Pile

- Pila 14

Code

- Coda 15

[2.][pdf-2] Alberi

Introduzione

Definizioni

- Albero radicato 16

Terminologia

- Profondità ¹⁷
- Livello ¹⁸
- Altezza ¹⁹

Alberi binari

Introduzione

- Albero binario ²⁰
- Albero radicato ²¹

Alberi strutturalmente diversi

- Alberi binari strutturalmente diversi ²²

[3.][pdf-3] Alberi binari di ricerca

Operazioni ABR

Successore / Predecessore

- Tree successorNode(Tree T) ²³

ABR bilanciati

Definizioni

- Fattore di bilanciamento ²⁴

Alberi Red-Black

- Albero Red-Black ²⁵
- Altezza nera di un nodo v ²⁶
- Altezza nera di un albero Red-Black ²⁷

[4.][pdf-4] Grafi

Introduzione

Definizioni

- Grafo orientato / Directed 28
- Grafo non orientato / Undirected 29
- Grafo pesato 30
- Grado di un nodo 31
- Cammino 32
- Cammino semplice 33
- Ciclo 34
- Ciclo semplice 35

Casi speciali

- Grafo completo 36
- Grafo ciclico 37
- Grafo aciclico 38
- Grafo orientato aciclico / DAG 39
- Albero libero 40
- Albero radicato 41
- Foresta 42
- Grafo trasposto 43

Componenti connesse

- Sottografo ⁴⁴
- Sottografo massimale ⁴⁵
- Raggiungibilità ⁴⁶
- Grafo connesso ⁴⁷
- Grafo fortemente connesso ⁴⁸
- Componente connessa ⁴⁹
- Componente fortemente connessa ⁵⁰

Alberi di copertura

- Albero di copertura ⁵¹
- Albero dei cammini minimi ⁵²

Visite dei grafi

- Visita per livelli ⁵³
- Visita ricorsiva ⁵⁴

DFS - DAG

Classificazione degli archi

- Arco dell'albero di copertura DFS ⁵⁵

Ordinamento topologico

- Ordinamento topologico ⁵⁶

Componenti fortemente connesse

Grafo delle componenti

- Grafo delle componenti ⁵⁷

[5.][pdf-5] Hashing

Introduzione

Definizioni

- Insieme universo \mathcal{U} ⁵⁸
- Funzione hash ⁵⁹
- Tabella hash ⁶⁰
- Collisione ⁶¹

Tabelle ad accesso diretto

- Tabelle ad accesso diretto ⁶²

Funzioni hash

Funzioni hash perfette

- Funzione hash perfetta ⁶³

Uniformità semplice

- Uniformità semplice ⁶⁴

Gestione delle collisioni

Indirizzamento aperto

Definizioni

- Ispezione ⁶⁵
- Estensione della funzione hash ⁶⁶
- Sequenza di ispezione ⁶⁷

Tecniche di ispezione

- Hashing uniforme ⁶⁸

[7.][pdf-7] Code con priorità

Definizione

- Priority queue ⁶⁹

Vettore heap

Heap

- Heap ⁷⁰

Alberi binari

- Albero binario perfetto ⁷¹
- Albero binario completo ⁷²

Alberi binari heap

- Albero binario max-heap (min-heap) ⁷³

[8.][pdf-7] Insiemi disgiunti

Introduzione

Definizione

- Collezione di insiemi dinamici disgiunti ⁷⁴

Rappresentante

- Rappresentante ⁷⁵

Euristiche

Alberi: euristica sul rango

- Rango ⁷⁶
-

Contenuti

- Dato ¹
- Tipo di dato primitivo ²
- Tipo di dato astratto ³
- Specifica di un TDA ⁴
- Implementazione di un TDA ⁵
- Struttura di dati ⁶
- Sequenza ⁷
- Insieme ⁸
- Dizionario ⁹
- Albero ordinato ¹⁰
- Grafo ¹¹
- Lista concatenata ¹²
- Sequenza di nodi ¹³
- Pila ¹⁴
- Coda ¹⁵
- Albero radicato ¹⁶
- Profondità ¹⁷
- Livello ¹⁸

- Altezza ¹⁹
- Albero binario ²⁰
- Albero radicato ²¹
- Alberi binari strutturalmente diversi ²²
- `Tree successorNode(Tree T)` ²³
- Fattore di bilanciamento ²⁴
- Albero Red-Black ²⁵
- Altezza nera di un nodo v ²⁶
- Altezza nera di un albero Red-Black ²⁷
- Grafo orientato / Directed ²⁸
- Grafo non orientato / Undirected ²⁹
- Grafo pesato ³⁰
- Grado di un nodo ³¹
- Cammino ³²
- Cammino semplice ³³
- Ciclo ³⁴
- Ciclo semplice ³⁵
- Grafo completo ³⁶
- Grafo ciclico ³⁷
- Grafo aciclico ³⁸
- Grafo orientato aciclico / DAG ³⁹
- Albero libero ⁴⁰
- Albero radicato ⁴¹

- Foresta ⁴²
- Grafo trasposto ⁴³
- Sottografo ⁴⁴
- Sottografo massimale ⁴⁵
- Raggiungibilità ⁴⁶
- Grafo connesso ⁴⁷
- Grafo fortemente connesso ⁴⁸
- Componente connessa ⁴⁹
- Componente fortemente connessa ⁵⁰
- Albero di copertura ⁵¹
- Albero dei cammini minimi ⁵²
- Visita per livelli ⁵³
- Visita ricorsiva ⁵⁴
- Arco dell'albero di copertura DFS ⁵⁵
- Ordinamento topologico ⁵⁶
- Grafo delle componenti ⁵⁷
- Insieme universo \mathcal{U} ⁵⁸
- Funzione hash ⁵⁹
- Tabella hash ⁶⁰
- Collisione ⁶¹
- Tabelle ad accesso diretto ⁶²
- Funzione hash perfetta ⁶³
- Uniformità semplice ⁶⁴

- Ispezione ⁶⁵
- Estensione della funzione hash ⁶⁶
- Sequenza di ispezione ⁶⁷
- Hashing uniforme ⁶⁸
- Priority queue ⁶⁹
- Heap ⁷⁰
- Albero binario perfetto ⁷¹
- Albero binario completo ⁷²
- Albero binario max-heap (min-heap) ⁷³
- Collezione di insiemi dinamici disgiunti ⁷⁴
- Rappresentante ⁷⁵
- Rango ⁷⁶

1. **Dato:** *valore che una variabile può assumere* ↔ ↔

2. **Tipo di dato primitivo:** *fornito direttamente dal linguaggio* ↔ ↔

3. **Tipo di dato astratto:** *modello matematico, definito da una collezione di valori e un insieme di operazioni ammesse su questi valori* ↔ ↔

4. **Specifica di un TDA:** *interfaccia che nasconde i dettagli implementativi* ↔ ↔

5. **Implementazione di un TDA:** *realizzazione vera e propria* ↔ ↔

6. **Struttura di dati:** *collezione di dati, caratterizzata dall'organizzazione della stessa (piuttosto che dal tipo dei dati contenuti)* ↔ ↔

7. Definizione (Sequenza)

Struttura dati **dinamica, lineare** che rappresenta una sequenza **ordinata** di valori che possono essere **ripetuti**

Operazioni

- Aggiungere/Togliere elementi (data la posizione)
- Accedere direttamente alla testa/coda
- Accedere sequenzialmente a tutti gli altri elementi ↔ ↔

8. Definizione (Insieme)

Struttura dati **dinamica, non lineare** che memorizza una collezione **non ordinata** di elementi **senza valori ripetuti**

Operazioni

- Operazioni base
 - Inserimento
 - Cancellazione
 - Verifica contenimento
- Operazioni insiemistiche
 - Unione

- Intersezione
- Differenza
- Operazioni di ordinamento
- Massimo
- Minimo
- Iteratori
- `foreach $x \in S$ do` ↔ ↔

9. Definizione (Dizionario)

Struttura dati che rappresenta il concetto matematico di **relazione univoca** o **associazione chiave-valore**

$$R : D \rightarrow C$$

- D : **chiavi**

- C : **valori**

Operazioni

- Ottenere il valore associato ad una particolare chiave o **nil** se assente
- Inserire una nuova associazione chiave-valore, cancellando eventuali associazioni precedenti per la stessa chiave
- Rimuovere un'associazione chiave-valore esistente ↔ ↔

10. Definizione (Albero ordinato)

Insieme finito di elementi detti **nodi**

- Uno di essi è designato come **radice**
- Gli eventuali nodi rimanenti sono partizionati in insiemi **ordinati** e **disgiunti**, anch'essi alberi ordinati ↔ ↔

11. Definizione (Grafo)

Struttura dati composta da

- Insieme di elementi detti **nodi** o **vertici**
- Insieme di coppie (ordinate oppure no) di nodi detti **archi** ↔ ↔

12. Definizione (Lista concatenata)



13. **Sequenza di nodi**, contenenti dati arbitrari e 1-2 **puntatori** all'elemento successivo e/o precedente

14. Definizione (Pila)

Struttura dati **dinamica**, **lineare** in cui l'elemento rimosso dall'operazione di cancellazione è predeterminato secondo il sistema **LIFO** (last-in first-out)

15. Definizione (Coda)

Struttura dati **dinamica**, **lineare** in cui l'elemento rimosso dall'operazione di cancellazione è predeterminato secondo il sistema **FIFO** (first-in first-out)

16. Definizione (Albero radicato)

Insieme di **nodi** e insieme di **archi orientati** che connettono coppie di nodi, con le seguenti proprietà

- Un nodo dell'albero è designato come **nodo radice**
- Ogni nodo n , a parte la radice, ha esattamente **un arco entrante**
- Esiste un **cammino unico** dalla radice ad ogni nodo
- L'albero è **connesso**

Definizione ricorsiva

Un albero è dato da (xor)

- Un **insieme vuoto**
- Un **nodo radice** e zero o più **sottoalberi**, ognuno dei quali è un albero
- La radice è connessa a radici di ogni sottoalbero con **archi orientati**

17. **Profondità**: *lunghezza del cammino semplice dalla radice a un nodo*

18. **Livello**: *insieme di nodi alla stessa profondità*

19. **Altezza**: *profondità massima della sue foglie*

20. Definizione (Albero binario)



21. **Albero radicato** in cui ogni nodo ha **al massimo due figli**, identificati come figlio sinistro e figlio destro ↔ ↔

22. Definizione (Alberi binari strutturalmente diversi)

Due alberi binari si dicono strutturalmente diversi se disegnando correttamente i figli destri e sinistri si ottengono figure diverse ↔ ↔

23. Definizione (Tree successorNode(Tree T))

Il **successore** di un nodo u è il **più piccolo nodo maggiore** di u

Restituisce

- Il **minimo** del sottoalbero **destro** di u , se presente
- Il **primo avo** v tale per cui u sta nel sottoalbero **sinistro** di v , altrimenti ↔ ↔

24. Definizione (Fattore di bilanciamento)

Il fattore di bilanciamento $\beta(v)$ di un nodo v è la **massima differenza** di altezza fra i sottoalberi di v ↔ ↔

25. Definizione (Albero Red-Black)

Albero binario di ricerca in cui:

- Ogni nodo è colorato di **rosso** o di **nero**
- Le **chiavi** vengono mantenute solo nei **nodi interni** dell'albero
- Le **foglie** sono costituite da nodi speciali Nil
- Vengono rispettati i seguenti **vincoli**
 1. La **radice** è nera
 2. Tutte le **foglie** sono nere
 3. Entrambi i **figli** di un nodo rosso sono neri
 4. Ogni **cammino** semplice da un nodo u ad una delle foglie contenute nel suo sottoalbero ha lo stesso numero di nodi neri ↔ ↔

26. Definizione (**Altezza nera di un nodo v**)

L'altezza nera $bh(v)$ di un nodo v è il numero di nodi neri lungo ogni cammino da v (escluso) ad ogni foglia (inclusa) del suo sottoalbero ↔ ↔

27. Definizione (**Altezza nera di un albero Red-Black**)

Altezza nera della sua radice ↔ ↔

28. Definizione (**Grafo orientato / Directed**)

Coppia $G = (V, E)$ dove

- V è un insieme di **nodi / vertici**
- E è un insieme di **coppie ordinate** (u, v) di nodi dette **archi**

↔ ↔

29. Definizione (**Grafo non orientato / Undirected**)

Coppia $G = (V, E)$ dove

- V è un insieme di **nodi / vertici**
- E è un insieme di **coppie non ordinate** (u, v) di nodi dette **archi** ↔ ↔

30. Definizione (**Grafo pesato**)

- Gli archi possono avere un **peso**
- Il peso $w(u, v)$ di un arco è dato da una **funzione di peso**

$$w : V \times V \rightarrow \mathbb{R}$$

- Se non esiste l'arco, il peso ha un valore ω che dipende dal problema ↔ ↔

31. Definizione (**Grado di un nodo**)

- **Grafi non orientati**

- Numero di archi incidenti su di esso

- **Grafi orientati**

- **Entrante:** numero di archi incidenti su di esso
- **Uscente:** numero di archi incidenti da esso ↔ ↔

32. Definizione (Cammino)

Un cammino C di lunghezza k è una **sequenza di nodi**

$$u_0, u_1, \dots, u_k \in V : (u_i, u_{i+1}) \in E, \forall i \in [0, k)$$



33. Definizione (Cammino semplice)

Cammino in cui tutti i suoi nodi sono distinti 

34. Definizione (Ciclo)

Un ciclo C di lunghezza $k > 2$ è una **sequenza di nodi**

$$u_0, u_1, \dots, u_k \in V : (u_i, u_{i+1}) \in E \wedge u_0 = u_k, \forall i \in [0, k)$$



35. Definizione (Ciclo semplice)

Ciclo in cui tutti i suoi nodi sono distinti, ad eccezione di primo e ultimo 

36. Definizione (Grafo completo)

Grafo con un arco fra tutte le coppie di nodi 

37. Definizione (Grafo ciclico)

Grafo che contiene almeno un ciclo



38. Definizione (Grafo aciclico)

Grafo non orientato che non contiene cicli



39. Definizione (Grafo orientato aciclico / DAG)

Grafo orientato che non contiene cicli ↔ ↔

40. Definizione (Albero libero)

Grafo connesso con $m = n - 1$

↔ ↔

41. Definizione (Albero radicato)

Albero libero nel quale uno dei nodi è designato come **radice/sorgente** ↔ ↔

42. Definizione (Foresta)

Grafo formato da un insieme di alberi ↔ ↔

43. Definizione (Grafo trasposto)

Dato un grafo orientato $G = (V, E)$, il grafo trasposto $G^T = (V, E_T)$ ha gli stessi nodi e gli archi orientati in **senso opposto**

$$E_T = \{(u, v) \mid (v, u) \in E\} \quad \leftrightarrow \quad \leftrightarrow$$

44. Definizione (Sottografo)

G' è un sottografo di G se e solo se

$$G' \subseteq G \iff V' \subseteq V \wedge E' \subseteq E$$

↔ ↔

45. Definizione (Sottografo massimale)

G' è un sottografo massimale di G se e solo se

$$G'_{\max} \subseteq G \iff \nexists G'' \subseteq G : G''_{\text{connesso}} \wedge G' \subset G'' \quad \leftrightarrow \quad \leftrightarrow$$

46. Definizione (Raggiungibilità)

Un nodo v è raggiungibile da u se esiste almeno un **cammino** da u a v ↔ ↔

47. Definizione (Grafo connesso)

Un grafo non orientato $G = (V, E)$ è connesso se e solo se ogni suo nodo è **raggiungibile** da ogni altro suo nodo



48. Definizione (Grafo fortemente connesso)

Come sopra, con un grafo orientato 

49. Definizione (Componente connessa)

Un grafo non orientato $G' = (V', E')$ è una componente connessa di G se e solo se G' è un sottografo connesso e massimale di G



50. Definizione (Componente fortemente connessa)

Come sopra, con un grafo orientato 

51. Definizione (Albero di copertura)

Dato un grafo $G = (V, E)$ non orientato e connesso, un albero di copertura di G è un sottografo $T = (V, E_T)$ tale che

- T è un albero

- $E_T \subseteq E$

- T contiene tutti i vertici di G 

52. Definizione (Albero dei cammini minimi)



Albero di copertura radicato in s avente un cammino da s a tutti i nodi raggiungibili da s 

53. **Visita per livelli:** prima la radice, poi i nodi a distanza i da essa

Soluzione (Visita in profondità / DFS)



54. **Visita ricorsiva:** per ogni nodo si visitano tale nodo e i suoi nodi adiacenti  

55. **Arco dell'albero di copertura DFS:** *arco esaminato da un nodo marcato ad un nodo non marcato*  

56. Definizione (Ordinamento topologico)

Ordinamento lineare dei nodi di un DAG G , tale per cui

$$(u, v) \in E \Rightarrow u < v \quad \text{↔} \quad \text{↔}$$

57. Definizione (Grafo delle componenti)

$$C(G) = (V_c, E_c)$$

$$- V_c = \{C_1, C_2, \dots, C_k\}$$

- C_i è la i -esima SCC di G

$$- E_c = \{(C_i, C_j) \mid \exists (u_i, u_j) \in E : u_i \in C_i \wedge u_j \in C_j\}$$

Proprietà

- C è aciclico

$$- C(G^T) = [C(G)]^T$$

- dt e ft di C corrispondono a quelli del primo nodo visitato in C

$$- dt(C) = \min \{dt(u) \mid u \in C\}$$

$$- ft(C) = \max \{ft(u) \mid u \in C\} \quad \text{↔} \quad \text{↔}$$

58. **Insieme universo \mathcal{U} :** *insieme delle possibili chiavi, di dimensione $\#\mathcal{U} = u$*  

59. **Funzione hash:** *funzione H che mappa chiavi $k \in \mathcal{U}$ in interi $H(k)$*  

60. Definizione (Tabella hash)

Vettore $T[0, \dots, m-1]$ di dimensione m , in cui

- Gli elementi di T appartengono all'insieme \mathcal{U}

- La posizione degli elementi è determinata da una funzione hash

$$H : \mathcal{U} \rightarrow \{0, 1, \dots, m-1\} \leftrightarrow \leftrightarrow$$

61. **Collisione:** *due o più chiavi nel dizionario mappate nello stesso valore hash* $\leftrightarrow \leftrightarrow$

62. Definizione (**Tabelle ad accesso diretto**)

Caso particolare in cui \mathcal{U} è un **sottoinsieme** (piccolo) di \mathbb{Z}^+

- Si utilizza la funzione hash identità $H(k) = k$

- Si sceglie un valore $m = u$ $\leftrightarrow \leftrightarrow$

63. Definizione (**Funzione hash perfetta**)

Funzione hash **iniettiva**

$$\forall k_1, k_2 \in \mathcal{U} : k_1 \neq k_2 \Rightarrow H(k_1) \neq H(k_2) \leftrightarrow \leftrightarrow$$

64. Definizione (**Uniformità semplice**)

- Sia $P(k)$ la probabilità che una chiave k *possa apparire* nella tabella

- Sia $Q(i)$ la probabilità che una chiave finisca nella cella i

$$Q(i) = \sum_{k \in \mathcal{U} : H(k)=i} P(k)$$

Una funzione hash H gode di uniformità semplice se

$$i \in \{0, \dots, m-1\}, \quad Q(i) = \frac{1}{m} \leftrightarrow \leftrightarrow$$

65. **Ispezione:** *esame di uno slot durante la ricerca* $\leftrightarrow \leftrightarrow$

66. Definizione (**Estensione della funzione hash**)

$$H : \underbrace{\mathcal{U} \times [0, \dots, m-1]}_{\text{Numero ispezione}} \rightarrow \underbrace{[0, \dots, m-1]}_{\text{Indice vettore}} \leftrightarrow \leftrightarrow$$

67. Definizione (**Sequenza di ispezione**)

Una sequenza di ispezione

$$[H(k, 0), H(k, 1), \dots, H(k, m-1)]$$

è una **permutazione degli indici** $[0, \dots, m-1]$ corrispondente all'ordine in cui vengono esaminati gli slot $\leftrightarrow \leftrightarrow$

68. Definizione (Hashing uniforme)

Situazione ideale in cui ogni chiave ha la stessa probabilità di avere come sequenza di ispezione una qualsiasi delle $m!$ permutazioni di $[0, \dots, m - 1]$ ↔ ↔

69. Definizione (Priority queue)

Struttura dati astratta, simile ad una coda, in cui ogni elemento inserito possiede una sua priorità

- **Min-priority queue**: estrazione per valori crescenti di priorità
- **Max-priority queue**: estrazione per valori decrescenti di priorità

Operazioni

- Inserimento in coda
- Estrazione dell'elemento con priorità di valore min/max
- Modifica priorità (decremento/incremento) di un elemento inserito ↔ ↔

70. Definizione (Heap)

Struttura dati speciale che associa i **vantaggi** di

- **Alberi**: esecuzione in tempo $O(\log n)$
- **Vettori**: memorizzazione efficiente ↔ ↔

71. Definizione (Albero binario perfetto)

- Tutte le foglie hanno la stessa profondità h
- Tutti i nodi interni hanno tutti grado 2
- Dato il numero di nodi n , ha altezza $h = \lfloor \log n \rfloor$
- Data l'altezza h , possiede $n = 2^{h+1} - 1$ nodi ↔ ↔

72. Definizione (Albero binario completo)

- Tutte le foglie hanno la stessa profondità h o $h - 1$
- Tutti i nodi a livello h sono 'accatastati' a sinistra
- Tutti i nodi interni hanno grado 2, eccetto al più uno
- Dato il numero di nodi n , ha altezza $h = \lfloor \log n \rfloor$ ↔ ↔

73. Definizione (Albero binario max-heap (min-heap))

Albero binario completo tale che il valore memorizzato in ogni nodo è maggiore (minore) dei valori memorizzati nei suoi figli

Proprietà

- Non impone una relazione di ordinamento totale fra i figli di un nodo
- È un **ordinamento parziale**
- Riflessivo: $n \geq n$
- Antisimmetrico: $n \geq m \wedge m \geq n \Rightarrow n = m$
- Transitivo: $n \geq m \wedge m \geq r \Rightarrow n \geq r$ ↔ ↔

74. Definizione (Collezione di insiemi dinamici disgiunti)

Collezione $S = \{S_1, S_2, \dots, S_k\}$ tale per cui

- $\forall i, j : i \neq j \Rightarrow S_i \cap S_j = \emptyset$
- $\bigcup_{i=1}^k S_i = S$, dove $k = |S|$ ↔ ↔

75. Definizione (Rappresentante)

Il rappresentante dell'insieme S_i è un suo qualunque membro che lo identifica univocamente

↔ ↔

76. Definizione (Rango)

Il rango $\text{rank}[x]$ di un nodo x è il numero di archi del cammino più lungo fra x e una foglia sua discendente

- Rango \equiv altezza del sottoalbero associato al nodo ↔ ↔