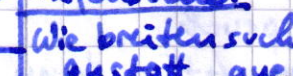
Breitensuche:

- alle Kn. undiscovered
- Startknoten disc. und in queue
- ↳ • kn. dequeuen und auf Suchgröße prüfen
  - • undisc. Nachbarknoten zu queue hinzuf. und disc.

Tiefensuche:

wie Breitensuche, bloß auf stack anstatt queue



Topsort:

$G=(V,E)$ mit $(u,v)\in E \quad \forall\, t(u)<t(v)$

kein Topsort für Kreis/Schlinge

- alle Kn. weiß
- in Tiefensuche besuchte weiße grau setzen.
  - ↳ wenn auf grau treffend: Fehler! (Kreis)
- bei Rekursion der Tiefensuche Knoten schwarz markieren und an Liste anfügen

↳ Liste rückwärts ist topsort

LZ Tiefen-/Breitensuche/Topsort: $\boxed{O(|V|+|E|)}$

Lineare Suche: LZ: $\boxed{O(n)}$

Voraussetzung: sortiertes Array

- in der Mitte starten (abrunden)
- wenn gesuchte Zahl kleiner als gewählte ist, linkes Teilarray auf gleiche Weise untersuchen. Sonst rechtes.

Suchbaum:

↳ binärer Wurzelbaum

- linker Teilbaum enthält kleinere Knoten
- rechter Teilbaum größere

↳ suche über Folgen der Pfade (Art binäre Suche)

LZ: $\boxed{O(\log(n))}$ vollst. bal.
$\boxed{O(n)}$ linear entartet

Hashing:

Abbildung Schlüssel → pos. in Array

$h(s) = s \bmod m$ → größe der Hashtabelle

bei Überlauf: Überlaufliste: pos. in Array hat Liste mit allen entspr. Hashelem.

im günstigen Fall: $\boxed{O(1)}$

Quicksort:

- 1. elem. markieren (Pivot)
  - ↳ alle elem., die kl. sind als Pivot links einsortieren, die größer sind rechts
- ↳ rekursiver Aufruf quicksort über linke/rechte Teillisten

⇒ nicht gut bei linear entarteten Listen

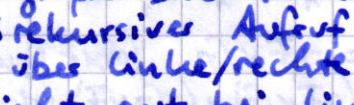
ØLZ: $\boxed{O(n \log n)}$ worst-case: $\boxed{O(n^2)}$

Mergesort:

- Liste halbieren (abgerundet)

immer weiter halb. bis nur noch 1-elem. Listen

⇒ 1-elem. Listen in Rekursion zusammenführen & dabei sortieren



LZ: $\boxed{O(n \log(n))}$ [$O(n)$ auf jeder Ebene]

Codierung: inv. Elem.: $\boxed{a+a^{-1}=0}$

Bsp.: $(\mathbb{Z}_5,+)$ ↳ inv.: $0+0=0$, $1+4=0$

Parität:

$b_n = \left(\sum_{i=1}^{n-1} b_i\right) \bmod 2$ aus Folge $b_1 \dots b_{n-1}$

↳ Codewort $b_1 \dots b_n$

$\boxed{\sum_{i=1}^{n} b_i \equiv 0 \pmod 2}$ Parity-Check-Code (PCC)

Mächtigkeit aller PCCs der Länge n: $\boxed{|P_n| = 2^{n-1}}$ [alle Bits bis auf das letzte können frei gewählt werden]

PCC: 1-fehlererkennend

Rekonstruktion von fehlendem Bit mögl.

ISBN: 10-stellig: 9 Buchnr. + 1 Prüfziffer

Prüfziffer: $z_{10} = \left(\sum_{i=1}^{9} i \cdot z_i\right) \bmod 11$

bzw. $\boxed{0 = \sum_{i=1}^{10} i \cdot z_i \pmod{11}}$

ISBN: 1-fehlererkennend, erkennt Zahlendreher

Fehlerkorrektur:

(Hamming-)Abstand: $d(u,v)$ ↳ Anzahl von Stellen, an denen sich u & v unterscheid.

Minimalabstand von Code C:

$\boxed{d(C) = \min\{d(u,v) \mid u,v \in C,\ u \neq v\}}$

bei lin. Code: • $d(u,v) = \sum d(u_i,v_i) = d(u-v,0)$
• $d(C) = \min\{d(c,0) \mid c \in C - \{0\}\}$

bei PCC: $d(PCC) = 2$

$d(C) \geq k+1 \Rightarrow$ k-fehlererkennend

$d(C) \geq 2k+1 \Rightarrow$ k-Fehler korrigierend

Linearer Code:

Parity-Check-Matrix

- C ist linear, wenn für Matrix $A = (\underbrace{1 \dots 1}_{n})$ gilt: $\boxed{C = \{w \mid A \cdot w^T = 0\}}$
- lin. Code C ist Vektorraum über $\mathbb{Z}_2(+,\cdot)$

Matrix G, deren Zeilen eine Basis eines lin. Codes C bilden heißt Generatormatrix

Umrechnung: Nachricht · G = Codewort

$A \cdot \text{Codewort}^T = 0$

Bsp.: $G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ PCC der Länge 4

↳ aus $(b_1\ b_2\ b_3) \cdot G$ wird $C = \{(b_1\ b_2\ b_3\ \sum_{i=1}^{3} b_i) \mid \dots\}$

Hamming-Codes per Rechteck-Code

Bsp.: $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$ Nachricht

Vektorraum: lin. unabh. Vektoren, mit denen sich alle Pkte erzeugen lassen

Basis: Menge lin. unabh. Vektoren, aller einschließen

Perfekter Code: (disjunkte Mengen, die alles einschließen)

PCC: wenn $v_1 \in C_1$ dann auch $v_1 + v_2 \in C$

Menge der Wörter, die zu 000 dekodiert werden: $S(000) = \{000, 100, 010, 001\}$

beim Code der Länge n mit $d(C) = k$

↳ $|S(v)| = \sum_{i=0}^{k} \binom{n}{i}$

↳ $S(v) = \{w \mid d(v,w) \leq k\}$

Logik:

Atom ≙ Aussage(-variable)

F	G	$\neg F$	$F \vee G$	$F \wedge G$	$F \to G$	$F \leftrightarrow G$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

$F \to G = \neg F \vee G$ (dort 0, wo $\neg F G$ [01], sonst 1)

$F \leftrightarrow G = (\neg F \vee G) \wedge (\neg G \vee F)$ (1 bei Gleichheit F & G)

wenn F, dann G: $F \to G$ (Od.: G, wenn F)

nur wenn F, dann auch G: $F \leftrightarrow G$

erfüllbar: es gibt gültige Belegung

unerfüllbar: es gibt keine —"— [⊥]

Tautologie: immer gültig [T]

$(p \Rightarrow q) \equiv (\bar{p} \Rightarrow \bar{q})$

$T \equiv p \wedge (p \Rightarrow q) \Rightarrow q$ (dir. Beweis)

$T \equiv p \wedge (\bar{q} \Rightarrow \bar{p}) \Rightarrow q$ (indir. Beweis)

$n! \leq n^n$

$\log(n!) \leq \log(n^n)$

bin $f(n) \leq g(n)$ gilt: $O(f) \leq O(g)$

$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

$\sum_{k=0}^{n} \binom{n}{k} = 2^n$

Kombinatorik:

Potenzmenge: Menge aller Teilmengen

Bsp.: $\mathcal{P}(\{1,2\}) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$ $\boxed{|\mathcal{P}(M)| = 2^{|M|}}$

Mächtigkeit: $|A|$ $\boxed{|A^n| = |A|^n}$

Bsp.: $A = \{0,\dots,9\}$, 6-stellige PIN → Alle PINs: $|A^6| = |A|^6 = 10^6$

Kombination:

Anzahl k-elem. Teilmengen aus n-elem. Menge: $\boxed{\binom{n}{k} = \frac{n!}{(n-k)!\,k!}}$ $\boxed{\binom{n}{k} = \binom{n}{n-k}}$

Bsp.: • Lotto 6 aus 49 ⇒ $\binom{49}{6}$

• (vollst.) Graph (binär → 2) aus n Knoten ⇒ $\binom{n}{2}$ Kanten

• Anzahl Graphen mit n Knoten, m Kanten ⇒: $\binom{\binom{n}{2}}{m}$

[Falls Elemente doppelt / mit sich selbst kombiniert werden dürfen: $\binom{n+k-1}{k}$]

zus. Graph: $(M \geq 3)$ Pfad der Länge 2 existiert

Permutation:

Perm. von n Zahlen: $\boxed{n!}$

Bsp.: Anz. Anordnungen einer 3-elem. Menge: 3!

[Falls Elem. mehrfach vorkommen dürfen: $\frac{n!}{n_1! \cdot \ldots \cdot n_k!}$]

Bsp.: 6 Sitzplätze, 3 Studenten

↳ $n = 6$ $n_{1,2,3} = 1$ (jeweils einzigartige Stud.)

$n_4 = 3!$ (immer 3 leere Sitzplätze)

↳ also $\frac{6!}{3!} = 6 \cdot 5 \cdot 4$ (1. Stud. hat 6 Auswahlmögl.; 2. 5; 3. 4)

k-te Ebene eines bal. WB hat 2^k Knoten

k-Permutation:

k-Tupel aus n-elem. Menge

für keine doppelten elem.: $\binom{n}{k} k!$

Bsp.: Wörter mit 4 Buchstaben (26) [ohne Wdh.]: $\binom{26}{4} \cdot 4!$

mit doppelten elem.: $\boxed{n^k}$

Bsp.: 8-stellige binäre Zahl Möglichkeiten: $2^8 = 256$

$\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$

bzw. $\sum_{k=0}^{n} x^k = \frac{1 - x^{n+1}}{1 - x}$

Graph (ungerichtet):

$G = (V, E)$ $V = \{1, \dots, n\}$ (Knoten) $E = \{\{1,1\}, \{1,2\}, \dots, \{n,n\}\}$ (Kanten)

Kanten (ungerichtet): ungeordnete Paare $\{u, v\}$

Vollständig: alle Knoten paarweise verbunden

↳ bei n Knoten besitzt vollst. Gr. $\boxed{\binom{n}{2}}$ Kanten (mit jeweils Kantengrad $\boxed{n-1}$)

Kantengrad: Anz. verbundener Kanten

$\boxed{\deg(v) = k}$ (Knoten v ist mit k anderen Knoten verbunden)

jeder Graph: $\boxed{\sum \deg(v) = 2|E|}$ Bsp.: n Knoten, k Kanten ↳ $n \cdot k = 2|E|$

Weg: Folge von Knoten $v_1, \dots, v_k$ $(v_i \in E)$

Länge Weg: $\boxed{k-1}$ Kreis, wenn: $\boxed{v_1 = v_k}$

Pfad: Weg ohne Mehrfachknoten

Beweis:

Direkt: $A \Rightarrow B$

Bsp.: Wenn a gerade, dann auch a^2

• $a = 2n$ (Def. von Gerade)

• $a^2 = 4n^2 = 2 \cdot 2n^2 \Rightarrow a^2$ gerade

Indirekt: $\neg A \Rightarrow \neg B$

Bsp.: Wenn a^2 gerade, dann auch a

• a ungerade: $a = 2n - 1$

• $a^2 = \underbrace{4(n^2 - n)}_{\text{gerade}} + \underbrace{1}_{\text{unger.}}$ $\Rightarrow a^2$ ungerade

Widerspruch:

Wenn $A \to T$, dann $\neg A \to \perp$

Bsp.: $\sqrt{2}$ irrational? ↳ $\sqrt{2}$ rational $\to \perp$

• Annahme $\sqrt{2}$ rational, also $2 = \frac{m^2}{n^2}$

• 2 ist Teiler von n und m, aber m und n sind Teilerfremd ↯

Vollständige Induktion:

„für alle $n \in \mathbb{N}$ gilt $A(n)$"

• IA: $n = 0$ $A(0)$ ist wahr!

• IV: $A(n)$ ist wahr!

• IS: $A(n) \Rightarrow A(n+1)$

O-Notation:

Pot./Log.-Gesetze:

$a^{x+y} = a^x \cdot a^y$ $\log(x \cdot y) = \log x + \log y$

$(a^x)^r = a^{x \cdot r}$ $\log(\frac{1}{x}) = -\log x$

$a^0 = 1$ $\log 1 = 0$

$\log x^r = r \cdot \log x$

$\log_a x = \frac{\log_b x}{\log_b a}$

Für Funkt. $f > 0$ ist $O(f)$ Menge der Funkt. g mit $g(n) \leq c \cdot f(n)$

Bsp.: • $2n^3 - n + 5 \leq 2n^3 + 5 \leq 7n^3 \in O(n^3)$

kl. 0, also weg lassen

• $2 \log(n^2 + 1) \leq 2 \cdot \log(n^3) = 6 \log(n) \in O(\log(n))$

$\boxed{n^2 + 1 \leq n^3}$ ↳ $\frac{1}{n} + \frac{1}{n^3} \leq 1 \Rightarrow 0 \leq 1$ ✓

• $n! \in O(n^n)$ • $\binom{n}{2} \in O(n^2)$

• $\log n! \in O(n \log n)$ • $2^{2n-1} \in O(4^n)$

zusammenhängend, wenn es von jedem Knoten Weg zu beliebigen anderen Knoten gibt.

Baum: zusammenhängend, keine Kreise, $\boxed{|E| = |V| - 1}$

Blatt: Knoten v mit $\boxed{\deg(v) \leq 1}$

Wurzelbaum: Baum mit def. Wurzelknoten

Binärer Wurzelbaum: jeder Knoten hat genau 2 Nachfolger

wenn bin WB balanciert:

• Pfad Wurzel → Blatt immer gleich groß

• bei Tiefe d: $\boxed{2^d}$ Blätter

↳ also: $\boxed{d = \log_2(n)}$ n = Blätter

Bin. WB: l: kürzestes Pfad von Knoten zu Blatt

$\Rightarrow \boxed{|V| \geq 2^{l+1} - 1}$