Cardinality Rules

Tuesday, March 30, 2021 11:00 AM

Product Rule:

For any sets A, B: $|A \times B| = |A||B|$

Sum Rule:

For any sets A, B: $|A \cup B| = |A| + |B|$

Set Partition

Tuesday, March 30, 2021 12:08 PM

A *set partition* of a set S is a collection of subsets of S $(A_1, \ldots, A_k)$ such that $A_i \subseteq S$ for all i and $A_i \cap A_j = \emptyset$ for all $i \neq j$ and $A_1 \cup \ldots \cup A_k = S$.

Recurrence Relation

Tuesday, March 30, 2021 12:13 PM

A recurrence relation defines a sequence. It is an equation with one or more base cases.

The equation is called a recurrence and it describes an entry in terms of previous entries.

Ex: $DT(n) = DT(n-1) + DT(n-2)$
$DT(1) = 1 \quad DT(2) = 1$] Fibonacci sequence.

Characteristic Polynomial

Thursday, April 1, 2021 11:40 AM

For certain recurrences, you can write out the characteristic polynomial and solve for the roots

Ex $r^2 = r + 1$

$$r^2 - r - 1 = 0$$

$$r = \frac{1 \pm \sqrt{1 - 4(1)(-1)}}{2} = \frac{1 \pm \sqrt{5}}{2}$$

$$\varphi = \frac{1 + \sqrt{5}}{2} \qquad \overline{\varphi} = -\frac{1}{\varphi} = \frac{1 - \sqrt{5}}{2}$$

$$Fib(n) = A\varphi^n + B\overline{\varphi}^n$$

Fibonacci Nearest Integer

Thursday, April 1, 2021 11:48 AM

$$Fib(n) = \frac{1}{\sqrt{5}}\left(\varphi^{n+1} + \left(-\frac{1}{\varphi}\right)^{n+1}\right)$$

as $n \to \infty$, $\left(-\frac{1}{\varphi}\right) \to 0$

That means $Fib(n) \approx \frac{1}{\sqrt{5}}\varphi^{n+1}$

In fact $Fib(n) = \left\lfloor \frac{1}{\sqrt{5}}\varphi^{n+1} \right\rceil$

Other Well Known Recurrences

Thursday, April 1, 2021 11:50 AM

Factorial: $0! = 1$ $n! = n \cdot (n-1)!$ for $n \geq 1$

Summation: $S(1) = 0$ $S(n) = S(n-1) + (n-1)$ for $n \geq 1$

Sum of pow: $R(1) = 1$ $R(n) = R(n-1) + (r^{n-1})$ for $n \geq 2$

Geometric Sum

Thursday, April 1, 2021 12:07 PM

A geometric sum is a sum of powers.

$$\sum_{i=0}^{n-1} r^i$$

And has closed form:

$$\sum_{i=0}^{n-1} r^i = \frac{r^n - 1}{r - 1}$$

So for $r = 2$

$$\sum_{i=0}^{n-1} 2^i = \frac{2^n - 1}{1} = 2^n - 1$$

Solving a Recurrence

Thursday, April 1, 2021 12:07 PM

1. Guess and Check

   Start with small values of n and look for a pattern.
   Confirm your guess with a proof by induction

2. Unravel

   start with the general recurrence and keep replacing
   n with smaller values. Keep unraveling until the base case

3. Characteristic Polynomial.

   If the recursion is of a certain form, you can guess
   that the closed form is Cw^n and solve for
   w by finding the root of a polynomial.

Tower of Hanoi

Tuesday, April 6, 2021 11:05 AM

Recursive: $T(n) = 2T(n-1) + 1$

$T(1) = 1$

Closed Form: $T(n) = 2^n - 1$

Closed Form Equations

Thursday, April 1, 2021 11:47 AM

<u>Closed form</u>: Does not involve recurrance to evaluate.

For example $f(x) = x^2 + 2x + 1$

<u>Idea</u>: We can represent recurrance relations as closed form using the solving recurrence strategies.

Power Rule

Tuesday, April 6, 2021 11:40 AM

For any set A: $|\underbrace{A \times A \times A \times \dots \times A}_{n}| = |A|^n$

More generally:

- To count the number of strings of length n over the finite alphabet A.
- To count the number of n-length sequences usin only the numbers $\{0, 1, \dots b\}$
- To count the number of ways to distribute b distinct objects among n people

Principle of Inclusion Exclusion

Tuesday, April 6, 2021 12:00 PM

Given two non disjoint sets A, B ($A \cap B \neq \emptyset$):

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Given n non disjoint sets $A_1, A_2 \ldots A_3$ are finite:

$$|A_1 \cup A_2 \cup \ldots A_n| = \sum |A_i| - \sum |A_i \cap A_j| + \sum |A_i \cap A_j \cap A_k| \ldots$$
$$+ (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n|$$

Standard Permutation

Tuesday, April 6, 2021 11:57 AM

Number of ways all n distinct objects can be ordered such that each object appears once.

Note: Is the number of bijections from a finite set to itself.

Note: Same as r-Permutation of length n:

$$P(n) = n!$$

Permutations as Functions

Thursday, April 8, 2021 11:14 AM

Def A permutation can be represented as a bijection from a finite set to itself:

$$\sigma : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$$

nPr (r-Permutations)

Tuesday, April 6, 2021 11:53 AM

The number of ways to arrange r objects out of n is:

r-Permutations: $nPr = P(n, r) = \frac{n!}{(n-r)!}$

Convention: $0! = 1$

Permutation Notations

Thursday, April 8, 2021 11:18 AM

One-line notation:

5, 6, 2, 4, 1, 3

Two-line notation:

1	2	3	4	5	6
5	6	2	4	1	3

Function notation:

$\sigma(1) = 5$

$\sigma(2) = 6$

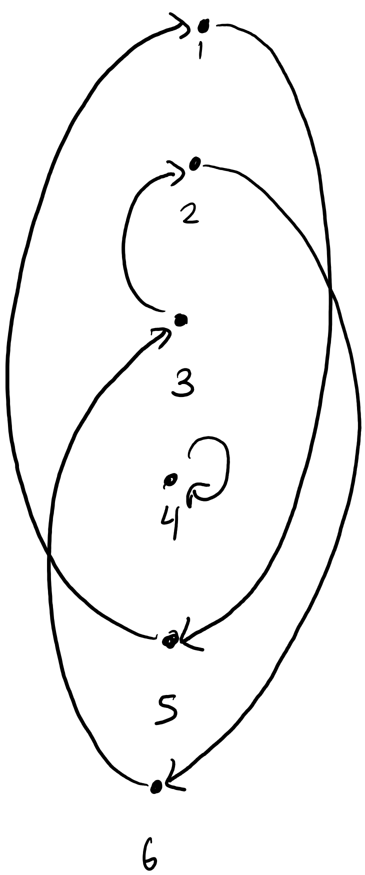
$\sigma(3) = 2$

$\sigma(4) = 4$

$\sigma(5) = 1$

$\sigma(6) = 3$

Directed Graph:



Cycle notation:

(1, 5), (2, 6, 3), (4)

nCr (r-combination)

Thursday, April 8, 2021 11:36 AM

<u>Def</u> An r-combination of r elements from a set of n distinct objects is an <u>unordered</u> selection (subset) of them:

$$nCr = C(n,r) = \binom{n}{r} = \frac{n!}{(n-r)!\,r!}$$

Uses of nCr

Thursday, April 8, 2021 12:10 PM

Def We claim that $\binom{n}{k}$ counts the number of

- k element subsets of a set of cardinality n
- Length n binary strings with exactly k ones

Binomial Theorem

Tuesday, April 13, 2021 11:01 AM

The value $\binom{n}{k}$ is also called the <u>binomial coefficient</u>.

<u>Thm</u> $(x+y)^n = \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \binom{n}{2}x^{n-2}y^2 \ldots$

$$= \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Binomial Coefficient Identities

Tuesday, April 13, 2021 11:16 AM

Thm Symetry Identity $\binom{n}{k} = \binom{n}{n-k}$

Thm Pascal's Identity $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$

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

Thm Hockey Stick Identity $\binom{n}{k} = \sum_{j=k}^{n} \binom{j-1}{k-1}$

Integer Equations

Thursday, April 15, 2021 11:03 AM

In general the equation $a_1 + a_2 + \ldots a_k = n$

where n and a_i's are non negative integers ($a_i \geq 0$)

# of stars = RHS constants = n

# of bars = # of variables $- 1 = k - 1$

There are $\binom{n+k-1}{k-1} = \binom{n+k-1}{n}$ solutions

If a_i's are positive integers ($a_i \geq 1$):

There are $\binom{n-1}{k-1}$ solutions.

12 Fold Way

Thursday, April 15, 2021 11:40 AM

Let's say that:

N is a set with $|N| = n$
X is a set with $|X| = x$
we want to put N into X

Elements of N	Elements of X	Any f	(Each X has at most 1 N) High bound	(Each X has at least 1 N) Low bound
Labeled	Labeled	X^N	$_xP_n$	Stirling Number
Unlabeled	Labeled	$\binom{N+X-1}{X-1}$	$\binom{X}{N}$	$\binom{N-1}{X-1}$
Labeled	Unlabeled	Stirling Numbers of the 2^{nd} Kind		
Unlabeled	Unlabeled	Integer partitions.		

Pigeonhole Principle

Thursday, April 15, 2021 11:55 AM

If there are p pigeons and h pigeonholes and $p > h$ then: one pigeonhole has at least 2 pigeons.

Proof (contrapositive):

If all pigeonholes have less than 2 pigeons then $p \leq h$

Suppose all pigeonholes have less than 2 pigeons. Then at most each hole has 1 pigeon. So, the number of pigeons can be at most h. Thus $p \leq h$.

Data Encoding

Tuesday, April 20, 2021 11:19 AM

Different kinds of data encoding:

- Lossy encoding: some of the data is lost in the process
- Error correcting: errors can be detected by the encoding
- Lossless encoding: the data can be restored without losing information

Lossy Compression, Single Value Decomposition

Tuesday, April 20, 2021 11:21 AM

<u>Idea</u> We can use linear algebra to compress data to a smaller size with minimal loss.

<u>Example</u> We can compress an image to smaller resolutions.

<u>Idea</u> In videos, we can store changes from previous frames rather than each frame.

Lossless Compression

Tuesday, April 20, 2021 11:26 AM

What does it mean for an encoding to be efficient?

- Length (size)
- Time to encode/decode

Ex To encode N characters of DNA:

ASCII

8 bits per character

8N bits to encode

A = 41

C = 43

T = 54

G = 47

Two-bit encoding

2 bits per character

2N bits to encode

A = 00

C = 01

T = 10

G = 11

Error Correcting

Tuesday, April 20, 2021 11:23 AM

Idea: We can design a system that can self-detect errors that may happen during transmission of data.
We can also design a system that can correct its own errors as well.

Fixed Length Encoding

Tuesday, April 20, 2021 11:53 AM

- In fixed length encoding, all symbols are using the same number of bits.
- When symbols are roughly evenly distributed, fixed length encoding works fine.

Strings

Tuesday, April 20, 2021 11:30 AM

<u>Def</u> In general for fixed-length encoding of a string over an alphabet Σ, you need a minimum of $\lceil \log_2 |\Sigma| \rceil$ bits per character.

Bits to encode characters in the finite alphabets:

Binary — $\lceil \log_2 (2) \rceil = 1$

Ternary — $\lceil \log_2 (3) \rceil = 2$

Quaternary — $\lceil \log_2 (4) \rceil = 2$

Decimal — $\lceil \log_2 (10) \rceil = 4$ (BCD)

Hex — $\lceil \log_2 (16) \rceil = 4$

Alphanumeric — $\lceil \log_2 (95) \rceil = 7$

Binary vs BCD

Tuesday, April 20, 2021 11:38 AM

In computers, integers are converted to base 2

In calculators, integers are converted to BCD

Variable Length Encoding

Tuesday, April 20, 2021 11:53 AM

When symbols occur more frequently than others, we can use shorter strings for those symbols. This is called <u>variable length</u> encoding.

Comma Free Encoding

Tuesday, April 20, 2021 12:03 PM

A comma free code is an encoding scheme in which no concatenation of two code words contains a valid code words word that overlaps both

Huffman Tree

Tuesday, April 20, 2021 12:05 PM

Building: Take the two least common symbols and put them as children of a tree, whose "value" is the sum of their frequencies. Repeat, taking the tree as a unified symbol. Assign the final tree structure 0's and 1's to find the encoding/decoding.

Zeckendorf's Theorem, Fibonacci Encoding

Thursday, April 22, 2021 8:41 PM

<u>Thm</u> Each positive integer can be uniquely written as the sum of non-consecutive fibonacci numbers:

<u>Ex</u> $99 = 89 + 8 + 2 \rightarrow 0100\ 100001_{fib}$

$47 = 34 + 13 \rightarrow 0000101_{fib}$

$375 = 233 + 89 + 34 + 13 + 5 + 1 \rightarrow 1001010101 01_{fib}$

<u>Fibonacci Encoding</u> We can use Zeckendorf's Theorem and write integers as a sum of non-consecutive Fibonacci numbers, and use a single 1 as a delimiter.

<u>Efficiency</u> You need around $1 + \log_{1.62} n$ bits to encode an integer n using Fib. encoding.

It is less efficient per number than base 2, but it is comma free.

Theoretical Optimal Encoding

Thursday, April 22, 2021 9:33 PM

A theoretically optimal encoding for a set of N different objects requires $\lceil \log_2 N \rceil$ bits.

A theoretically optimal encoding for length n binary strings with exactly k 1s would use $\lceil \log_2 \binom{n}{k} \rceil$ bits per string.

How?
- List all length n binary strings with exactly k 1s in some order
- To encode: Store the position of a string in the list
- To decode: Given a position in the list, find the string

Ranking/Unranking

Thursday, April 22, 2021 9:47 PM

Idea: Instead of listing all possibilities, if we order the bitstrings in a particular way, we can use a ranking and unranking algorithm that converts between bitstrings and positions.

Ranking: Bitstring → Position : Bitstring of length n, with k "1"s:

1) Assign each bit a number from n-1 to 0
2) Assign each 1 a number in decreasing order from k to 1
3) For each 1, add the $\binom{\text{first number}}{\text{second number}}$

Ex: s = 001011001

8 7 6 5 4 3 2 1 0

$\binom{6}{4} + \binom{4}{3} + \binom{3}{2} + \binom{0}{1} = 15 + 4 + 3 + 0 = 22$

Unranking: Position → Bitstring : Position of bitstring of length n, with k "1"s:

r = p

output = 0000.....0 } len n

while k > 0

  Find a largest x such that $\binom{x}{k} \leq r$

  Put a 1 in position x (counting from the right)

  $r = r - \binom{x}{k}$

  k = k - 1

Lexicographical Order

Thursday, April 22, 2021 9:50 PM

<u>Def</u> Lexicographical order is a way to order fixed density binary strings. We can compare two fixed density binary strings to easily tell which string comes befor the other in this ordering.

Sorting

Tuesday, April 27, 2021 10:59 AM

Why sort?

- Organize data
- As a subroutine for algorithms
- Make searching, finding max/mins, finding duplicates easier
- Introduce: iterative algorithms, recursion, divide and conquer
- Sort aids in ranking

How to sort?

- Moving and inserting
- Swap consecutive positions
- Swap any two positions
- Split the list and reorganize

Sorting Lists with Comparisons

Tuesday, April 27, 2021 11:36 AM

Given a list $a_1 \ldots a_n$ we want to arrange the values such that $a_1 \le a_2 \le \ldots \le a_n$

Selection Sort

- find the smallest element
- swap the min with the next unsorted element
- repeat with remaining values

Best Case: 0 swaps Worst Case: $n-1$ swaps

Sorting Permutations with Swaps

Tuesday, April 27, 2021 11:19 AM

Suppose we can only swap any two positions and we want to sort a permutation of size n

Best Case: permutation already in order 0 swaps

Worst Case: permutation in cyclical shift $n-1$ swaps

<u>How to sort</u>

Find the cycle order

Reverse the cycle order and write as swap pairs

ie. $(1, 8, 9, 7) = (7, 9) \ (9, 8) \ (8, 1)$

Repeat for each remaining cycle

<u>Thm</u> (steps required)

Let σ be a permutation of $\{1 \ldots n\}$

Let $\sigma = (C_1, C_2 \ldots C_k)$ be the cycle decomposition of σ

Then σ can be sorted using

$$\sum_{i=1}^{k} (|C_i| - 1) \text{ swaps.}$$

Sorting Lists with Consecutive Swaps

Tuesday, April 27, 2021 11:53 AM

Suppose we can only swap consecutive elements to sort a list $a_1, \dots, a_n$:

Bubble Sort

- Compare two consecutive elements
- Swap if they are out of order
- Continue to end of list
- Repeat the three steps above from the beginning reducing the end of the list by one each time until sorted.

Best Case: Worst Case: $\frac{n(n-1)}{2}$ swaps

Early Exit: we can exit early if there were no swaps for an entire list.

Best Case: 0 swaps Worst Case: $\frac{n(n-1)}{2}$ swaps

Selection Sort vs Bubble Sort

Tuesday, April 27, 2021 12:04 PM

Selection

comparisons: $\frac{n(n-1)}{2}$

swaps: $n-1$ worst
0 best

Bubble

comparisons: $\frac{n(n-1)}{2}$ worst $n-1$ best

swaps: $\frac{n(n-1)}{2}$ worst 0 best

Searching

Tuesday, April 27, 2021 12:07 PM

<u>Def</u> Searching is attempting to find an item in a collection. We can indicate if an item was found by giving its location or by indicating it was not found.

<u>Why?</u> Finding files

Finding errors in code

Searching the internet

Linear Search and Binary Search

Tuesday, April 27, 2021 12:09 PM

<u>Linear Search</u>

While there are more items in the list:

- look at the next item
- if it is the item we are looking for, return it
- continue through the list
- if we reached the end of the list without finding the item, then it does not exist

Best Case: 1 comparison Worst Case: n comparisons

<u>Binary Search</u>: The list must be sorted

- find the midpoint
- if the midpoint > target, search the lower half
- if the midpoint < target, search the higher half
- if the midpoint = target, we found the target
- repeat until there are no halves to search through

Best Case: 1 comparison Worst Case: $\lceil \log_2 (n+1) \rceil$ comparisons

Asymptotic Classes and Notation

Thursday, April 29, 2021 11:09 AM

$f(n) \in \Theta(g(n))$ means that:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c \quad \text{where } c \text{ is finite and } c \neq 0$$

$f(n) \in o(g(n))$ means that:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

$f(n) \in O(g(n))$ means that

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c \quad \text{where } c \text{ is finite}$$

$f(n) \in \Omega(g(n))$ means that

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c \quad \text{where } c > 0 \text{ or } c = \infty$$

Common Asymptotic Notation

Thursday, April 29, 2021 12:15 PM

c	constant
$\log(\log(n))$	double logarithm
$\log(n)$	logarithm
$\log(n)^k$	poly-logarithm
n	linear
$n\log(n)$	log-linear
n^k	polynomial
a^n	exponential
$n!$	factorial
n^n	

Finding Algorithm Bounds

Tuesday, May 4, 2021 11:23 AM

In some code:

```
while (condition):
    do something
```

we can count the number of steps by multiplying the number of times the loop runs by the number of steps the action inside the loop takes.

Loop Invariants

Tuesday, May 4, 2021 11:56 AM

A loop invariant is a property that remains true after each loop iteration.

For an iterative algorithm:

1) Look for a loop invariant
2) Prove that it is invariant
3) Use the invariant to prove correctness.

Proving Recurrence Algorithm Correctness

Thursday, May 6, 2021 11:15 AM

Overall structure: Prove the algorithm is correct on input size of n. Then prove for more than n

Base case(s): Same as base cases for the recursion

Use induction to prove.

Divide and Conquer General Strategy

Thursday, May 6, 2021 12:12 PM

- Divide the problem of size n into a subproblems of size $\frac{n}{b}$
- Recursively solve each subproblem
- Conquer the problem of size n by combining solutions

Master Theorem

Thursday, May 6, 2021 12:14 PM

Divide and conquer algorithms have recurrences of the form

$T(n) = a \cdot T(\frac{n}{b}) + g(n)$.

If $g(n)$ is a polynomial represented by $O(n^d)$:

<u>Master theorem</u>: $T(n)$ is the time to solve problem of size n

$T(1)$ is constant

$g(n)$ is the time to do the conquer step to solve problem of size n.

and $a > 0$ $b > 1$ $d \geq 0$

$$T(n) \in \begin{cases} O(n^d) & \text{if } a < b^d \\ O(n^d \log n) & \text{if } a = b^d \\ O(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Fast Multiplication Algorithms

Tuesday, May 11, 2021 11:06 AM

Recap: Multiplication: Given two n-digit integers

$a = a_{n-1} a_{n-2} \cdots a_1 a_0$

$b = b_{n-1} b_{n-2} \cdots b_1 b_0$

return some representation of their product

Gradeschool Method: Compute partial products, shift, then add

Runtime: $O(n^2)$

Divide and Conquer: Divide n-digit numbers into two $\frac{n}{2}$-digit numbers

If $a = 1234\ 5678$ $b = 2468\ 1357$

$a = (1234)\cdot 10^4 + (5678)$

$b = (2468)\cdot 10^4 + (1357)$

then $a \cdot b = [(1234)\cdot 10^4 + (5678)]\cdot[(2468)\cdot 10^4 + (1357)]$

$= (1234)(2468)\cdot 10^8 + (1234)(1357)\cdot 10^4 + (5678)(2468)\cdot 10^4 + (5678)(1357)$

Karatsuba Multiplication

Tuesday, May 11, 2021 11:26 AM

Karatsuba Multiplication:

$$T_K(n) = 3\,T_K\left(\frac{n}{2}\right) + pn$$

$$\in O\left(n^{\log_2 3}\right) = O\left(n^{1.58\ldots}\right)$$

Graphs

Tuesday, May 11, 2021 11:28 AM

Types of graphs

- Directed graph ; Arrows on edges, ordered pair of vertices (v, w)
- Undirected graph : No arrows on edges, unordered set of vertices $\{v, w\}$
- Multigraph : Can have multiple edges per pair of nodes
- Simple graph : Can not have self loops.

Adjacency Matrix : Uses $O(|V|^2)$ memory

Adjacency List : Uses $O(|E|)$ memory

Vertex Degrees

Tuesday, May 11, 2021 11:58 AM

Def The degree of a vertex in an undirected graph is the total number of edges incident with it, except a self loop counts twice.

Def In general, the sum of degrees of all vertices is $2 \cdot |E|$
Thus, the number of vertices with odd degree is even.

$$\sum \text{degrees} = 2 \cdot |E|$$

Def The degrees of a vertex are indegree and outdegree.
Indegree is the number of edges going into V
Outdegree is the number of edges going out of V
Indegree = Outdegree = $|E|$

Walk, Path, Circuit, Cycle

Tuesday, May 11, 2021 12:06 PM

Def Walk : describes a route from one vertex to another
$(v_1, e_1, v_2, e_2 \ldots v_k)$

Def Path : a walk that does not repeat edges
length of path is the number of edges
there is always a path of length 0 from a vertex to itself

Def Simple Path: a path that does not repeat vertices

Def Circuit : path that starts and ends at the same vertex, len > 0

Def Cycle : circuit that does not repeat vertices, len > 0

Def Loop : an edge from a vertex to itself.

Complete, Edgeless, Complement, Bipartite, Tree Graphs

Tuesday, May 11, 2021 12:10 PM

- Complete graph : K_n is an undirected graph with n vertices such that there is an edge between any two vertices
- Edgeless graph : $\overline{K_n}$ is an undirected graph with n vertices with no edges.
- Complement graph : The complement of graph G, $\overline{G}$ is the graph such that
  $\overline{G}$ has an edge between u and v iff G does not have an edge there for all u, v
- Bipartite graph : A graph with two sets of vertices V_1, V_2 such that no two vertices in the same set are connected
- Tree : A graph with no cycles.

Graphsearch DFS, BFS, Dijkstra

Monday, May 17, 2021 12:14 AM

- Given a directed graph G and start vertex s,
- produce a list of all vertices w reachable from s by a directed path in G.

Def Graph search:

we partition each vertice in G into:

X: explored

F: frontier (reached but not explored)

U: unreached

Each vertex starts in U, then can be moved into F and then moved into X.

$U \overset{?}{\to} F \to X$ some vertices stay in U forever

Procedure Initialize $X = \emptyset$, $F = \{s\}$, $U = V - F$ (everything else)

while F is not empty:

Pick a w in F

for each outgoing neighbor y of w: for every $(w, y) \in E$:

If y is not in X or F, then move y from $U \to F$

Move w from $F \to X$

Return X

DFS/BFS If we use a stack to hold F, it is DFS

If we use a heap/queue, it is BFS

If we use a priority queue, it is Dijkstra's

Runtime of Graphsearch

Monday, May 17, 2021 1:20 AM

$$\sum_{w \in V} c' + c \cdot \deg(w)$$

$$= c'|V| + c\,|E|$$

thus graphsearch is on order of : $O(|V| + |E|)$

Purposes of Graphsearch

Monday, May 17, 2021 1:22 AM

Questions that can be answered:

- Where can I reach from s?
- Is the graph connected?
- Is a certain edge a bridge?
- Is there a cycle in my graph
- What is the degree of each vertex?
- (Using a queue: BFS) What is the shortest distance from s to all other reachable vertices?

Undirected Graphs: Connected, Disconnected

Tuesday, May 18, 2021 11:06 AM

Def Two vertices u, v are <u>connected</u> if there is a path from u to v

Def An undirected graph is <u>connected</u> if for any pair of vertices v, w there exists a path from v to w.

Def An undirected graph is <u>disconnected</u> if there exists two vertices v, w such that there is not a path from v to w.

- Disconnected graphs can be broken into pieces where each piece is connected
- Each maximal connected piece of the graph is a <u>connected component</u>

Directed Graphs: Strongly, Weakly Connected

Tuesday, May 18, 2021 11:13 AM

Def A directed graph G is strongly connected if for any ordered pair of vertices (v, w) there is a directed path from v to w

Def A directed graph G is weakly connected if the underlying undirected graph is connected.

Note: Strongly connected implies weakly connected

Eulerian Path, Eulerian Circuit

Tuesday, May 18, 2021 11:20 AM

<u>Def</u> An <u>Eulerian path</u> is a <u>path</u> through a graph G such that each edge is travelled exactly once.

A <u>Eulerian circuit</u> is a <u>circuit</u> through a graph G such that each edge is travelled exactly once.

Finding Existence: Does graph G contain an Euler path?

G must be undirected, connected graph with n vertices

Every degree of intermediate vertice have even degree

Start and end vertices must have odd degree

- Circuits must have all vertices with even degree

Thus: <u>Thm</u> If G has an Eulerian path, G has <u>two</u> or <u>zero</u> odd degree vertices

Finding Path: Find an Euler path in graph G

Start at one of the odd degree vertices

Travel around the graph, avoiding <u>bridges</u>

↳ an edge, if removed, causes G to become disconnected

Continue until we reach the end.

Fleury's Algorithm

Tuesday, May 18, 2021 11:50 AM

Eulerian paths: HOW (Fleury's Algorithm)

1. Input is a connected undirected graph G
2. Start at vertex v, an odd-degree vertex if possible.
3. **While** there are still edges in G:
4. **If** there is more than one edge incident on v:
5. Check that G has at most 2 odd-degree vertices.
6. Cross any edge incident on v that is not a bridge
7. **Else:**
8. cross the only edge available from v and add edge to output.
9. Delete the edge just crossed from G, update v.

Existence of Euler Path in Directed Graph

Tuesday, May 18, 2021 12:11 PM

If v is intermediate

$$\text{indegree}(v) = \text{outdegree}(v)$$

If v is an initial vertex

$$1 + \text{indegree}(v) = \text{outdegree}(v)$$

If v is final vertex

$$\text{indegree}(v) = \text{outdegree}(v) + 1$$

Hamiltonian Path

Tuesday, May 18, 2021 11:28 AM

Def A Hamiltonian path is a path where each vertex occurs exactly once. Hamiltonian cycle is a path that starts and ends at the same vertex.

Finding Existence: Does graph G contain a Hamiltonian Path?

Finding Path: Find a Hamiltonian Path in graph G

Note: Hamiltonian path problems turn out to be NP-complete

Solving Problems Using Graphs

Tuesday, May 18, 2021 11:33 AM

- Model problem as a graph

  Choose a vertex set & edge set .... many possible options

- Use graph algorithms to solve problem

  Identify what type of graph problem is needed to solve it.

Trees

Thursday, May 20, 2021 11:05 AM

<u>Def</u> An undirected graph is called an unrooted tree if it is connected and has no cycles. This means there is exactly one simple path between every two vertices.

- <u>Leaves</u> in a tree are vertices of degree 1
- All <u>trees with n vertices</u> have <u>n-1 edges</u>
- A set of trees is a <u>forest</u>
- Between each <u>pair of vertices</u> has <u>1 simple path</u>
- Each edge is a bridge.

<u>Def</u> A rooted tree is a weakly connected directed acyclical graph which has a root vertex with no incoming edges, and every other vertex has exactly one incoming edge

- Every tree has a height, which is the maximum height of all vertices

Binary Trees

Thursday, May 20, 2021 11:36 AM

Def Binary trees are rooted trees such that every parent has at most two children.

The max height with n vertices: $n-1$

The min height with n vertices: $\log_2(h+1)$

DAG, Topological Ordering

Thursday, May 20, 2021 12:01 PM

Directed graphs with no cycles are called directed acyclic graphs (DAGs)

A topological ordering or linearization of a graph is an ordered list of all its vertices such that for each directed edge (v, w), v comes before w in the list.

Probability (Terms)

Thursday, May 20, 2021 12:11 PM

Def Sample Space : set of possible outcomes

Def Outcome : element of set

Def Event : subset of sample space

Def Distribution : function

Uniform Distribution

Thursday, May 27, 2021 12:06 AM

Def A uniform distribution on a set S is the distribution which assigns the same probability to every outcome.

Thus, each outcome has probability $\frac{1}{|S|}$

Def If a sample space S has the uniform distribution, then for any event the probability of E is:

$$P(E) = \frac{|E|}{|S|}$$

Non-Uniform Distribution

Thursday, May 27, 2021 12:18 AM

<u>Def</u> Any distribution that is not uniform is a non-uniform distribution. The sum of all probabilities in the sample space is still 1.

Odds vs Probability

Thursday, May 27, 2021 12:21 AM

__Def__ If you have a probability p, then you have:

$\frac{1}{p}$ for 1 odds or $\frac{1}{p-1}$ to 1 odds

Bernoulli Trial, Binomial Distribution

Thursday, May 27, 2021 12:22 AM

Def Bernoulli Trial: a performance of an experiment with two outcomes.

Def Binomial Distribution: probability of exactly k successes in n independent Bernoulli Trials, when probability of success is p.

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Conditional Probability

Thursday, May 27, 2021 12:27 AM

Probability of an event may chance if additional events / information are provided.

<u>Def</u>. Suppose A and B are events, and $P(B) > 0$. Then:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{thus} \quad P(A \cap B) = P(A|B) \cdot P(B)$$

<u>Note</u>: $P(A|B)$ means the probability of A <u>given</u> B.

Independent Events

Thursday, May 27, 2021 12:36 AM

<u>Def</u> Two events A and B are <u>indepent</u> if the occurrence of one event does not effect the liklihood of the other event. Thus:

$P(A|B) = P(A)$ and $P(A \cap B) = P(A)P(B)$

<u>Note</u> Two independent events are not nessasarily disjoint.

Bayes' Theorem

Thursday, May 27, 2021 12:42 AM

We know:

$$P(A \cap B) = P(A \mid B)\, P(B)$$

and $A \cap B = B \cap A$

thus $P(A \cap B) = P(B \cap A) = P(A \mid B)\, P(B)$

<u>Thm</u> $P(B \mid A) = \dfrac{P(A \mid B)\, P(B)}{P(A)}$

$$P(B \mid A) = \frac{P(A \mid B)\, P(B)}{P(A \mid B)\, P(B) + P(A \mid \bar{B})\, P(\bar{B})}$$

Random Variables, Expected Value

Thursday, May 27, 2021 12:56 AM

Def A random variable X assigns a real number value to each outcome of an experiment.

$X : S \rightarrow \mathbb{R}$

Def The distribution of a random variable X is the function

$r \longrightarrow P(X = r)$

Def The expectation (average, expected value) of a random variable X on a sample space S is:

$$E(X) = \sum_{s \in S} p(s) X(s)$$

$$= \sum_{r \in X(S)} P(X = r)\, r$$

Partitioning a Sample Space

Tuesday, June 1, 2021 12:07 AM

Def For any two events, A and B, you can calculate the probability of A by partitioning the sample space based on B and $\bar{B}$

$$P(A) = P(A|B)P(B) + P(A|\bar{B})P(\bar{B})$$

Def Let $\{B_1 \ldots B_n\}$ be any partition of the sample space S

(i.e. $B_i \cap B_j = \emptyset$ for all $i \neq j$ and $B_1 \cup \ldots \cup B_n = S$

Then for any event A:

$$P(A) = \sum_{i=1}^{n} P(A|B_i)P(B_i)$$

Conditional Expectation

Tuesday, June 1, 2021 12:27 AM

Def Let X be a random variable. Let A be an event.

$$E[X \mid A] = \frac{1}{P(A)} \cdot \sum_{a \in A} P(a) X(a)$$

Alternatively:

$$E[X \mid A] = \sum_{r \in X(s)} r P(X = r \mid A)$$

Def From above, if X is a random variable and A is an event:

$$E(X) = P(A) E(X \mid A) + P(\bar{A}) E(X \mid \bar{A})$$

Linearity of Expectation

Tuesday, June 1, 2021 12:56 AM

<u>Thm</u> If X_i are random variables on S with no special assumptions:

$$E(X_1 + \dots + X_n) = E(X_1) + \dots + E(X_n)$$

Independent Random Variable

Tuesday, June 1, 2021 11:12 AM

Def Two variables X, Y are independent if

$$P(X = v \text{ and } Y = u) = P(X = v) \cdot P(Y = u) \quad \text{for all possible } v, u$$

Thm For two independent random variables X, Y:

$$P(XY) = P(X)P(Y)$$

Variance, Law of Large Numbers

Tuesday, June 1, 2021 11:17 AM

Def Variance is a measure of how close things are. In probability, we are interested in the variance of the expected value.

Thm The variance of X is:

$$V(X) = E(|X-E|)^2 = E(X^2) - E(X)^2, \quad V(X) \geq 0$$

Thm If two random variables X, Y are independent:

$$V(XY) = V(X) + V(Y)$$

Def For any constant c,

$$V(cX) = c^2 V(X)$$

Def Law of Large Numbers:

Given n independent trials $X_1 \dots X_n$ and assume they have the same variance:

$$\overline{X}_n = \frac{1}{n}(X_1 + \dots + X_n)$$

Thus:

$$V(\overline{X}_n) = \frac{V(X_i)}{n}$$

As $n \to \infty$, $V(\overline{X}_n) \to 0$

This means that as you do more trials, the variance of the average tends to 0, which means that the average is less likely to be far away from the expected value.