

cs271 Exam 3 Review Guide

Topics

- Heaps
- Hash Tables
- Random Number Generators
- Compression
 - RLE
 - Huffman Codes
 - LZW
 - bzip2
 - jar
 - mpeg-1 & 2
- Infix, Prefix, Postfix
- Graphs
 - Prim's, Minimum Spanning Tree
 - Dijkstras
 - Coloring Problem
 - Hamiltonian
 - DAG

Specifically not on Exam 3

- CS Department discussion on last lecture

Sample Exam Questions:

- 1.) What is a nearly complete tree? And is (5 (4 (2) 3) (4 6)) a nearly complete tree?
- 2.) Build a max heap given these input numbers ... [6 5 3 8 4 7 1 3]
- 3.) Build a min heap given these input numbers ... [6 5 3 8 4 7 1 3]
- 4.) What is the run-time of the following heap operations?
 - Insert
 - Delete Max
- 5.) Explain in your own words how a random generator in C++ works.
- 6.) In a hash table, what is a collision?
- 7.) Give an example of a 2 hash functions where the key is a person's student id.
- 8.) Given a hash table of size 10, using linked list for collisions, and this simple hash function ...
int hash(int key) { (key + 3) % 10 ; }
Insert the following elements : 6, 4, 9, 51, 441, 23, 2, 7, 56, 23, 12

9.) List the pros and cons of using buckets for collision resolution in a hash table

10.) Explain the idea of linear probing for collision resolution in hash tables

11.) Given these characters and probabilities, build the Huffman tree.

A = 56%, B = 13%, C = 12%, D = 11%, E = 10%, F = 8%, G = 2%

12.) List out the actual Huffman codes for the characters mentioned in #11 above

A =

B =

C =

D =

E =

F =

G =

13.) Explain what interlaced video is in MPEG-2.

14.) Convert this prefix notation to infix ... $+/A*BCDE$

15.) Convert this infix notation to postfix ... $(A + B) / C * (D + E + (F / G))$

16.) Explain what Precedence is.

17.) Explain what Associativity is

18.) Draw your own undirected connected graph with random edge lengths, then step by step using the algorithm we did in class, compute the minimum spanning tree

19.) Draw your own directed graph, with random arc lengths, then pick 2 nodes, and step by step using the dijkstra's algorithm, compute the shortest path length.