COS 226
Final Exam Review
Fall 2015

Ananda Gunawardena

(guna)

guna@cs.princeton.edu

Logistics

- The final exam **time and location**
  - 7:30-10:30 PM on Friday, January 15. McCosh 50
  - The exam will start and end promptly, so please do arrive on time.
- **Exam Format**
  - Closed book, closed note.
  - You may bring one 8.5-by-11 sheet (both sides) with notes in your own handwriting to the exam.
  - No electronic devices (e.g., calculators, laptops, and cell phones).

What to focus on

- focus on understanding fundamentals, not memorizing details (eg: code)
- ***For each algorithm***
  - *Write down as many algorithms as you can recall*
  - understand how it works on typical, worst case, best case input
  - Why do we care about this algorithm?
  - How is it different from other algorithms for the same problem?
  - When is it effective?

What to focus on

- ***For each data structure***
  - *Write down as many data structures as you can recall*
  - understand how it works on typical input
  - Why do we care about this data structure?
  - What makes us choose one data structure over another?
  - When is it effective?

Material covered

- The exam will *stress* material covered since the midterm, including the following components.
  - Lectures 13–23.
  - *Algorithms in Java, 4th edition*, Chapters 4–6.
  - Exercises 12–22.
  - Programming assignments 6–8
    - Wordnet, seam-carving, burrows-wheeler

Topics covered

Depth-first search

Breadth-first search

Kruskal's algorithm

Dijkstra's algorithm

Key-indexed counting

LSD radix sort

Knuth-Morris-Pratt substring search

Boyer-Moore substring search

RE to NFA

R-way tries

Run-length coding

Huffman coding

Topological sort

Prim's algorithm

Bellman-Ford algorithm

Ford-Fulkerson algorithm

MSD radix sort

3-way radix quicksort

Rabin-Karp substring search

Ternary search tries

Reductions

LZW compression

Burrows-Wheeler

Wordnet

1. What data structures are used to store wordnet?

2. What data structures are used to store SCA?

3. Is there a reason that we used BFS not DFS?

4. What is the order of the best algorithm that can find the length of the common ancestor?

5. What is the order of the best algorithm that can find the common ancestor?

6. What is a rooted DAG and how do we determine that? Order or growth of your algorithm?

7. If the wordnet is NOT a rooted DAG, will answers to 3 and 4 will hold?

8. Given a list of n nouns, What is the order of growth of the outcast algorithm?

Seam-Carving

- What is the purpose of the seamcarving assignment?
- How does it relate to shortest path?

- How to find Vertical and Horizontal seams?

- Why do a defensive copy of Picture?

- What is the order of growth for the two methods, removeHorizontalSeam and removeVerticalSeam?

- Can seamcarver be extended to video scaling? How?

Burrows-Wheeler

- What libraries were used to read and write input/output to the program?
- What method in the output library that was required to print the output correctly?
- What data structures were used to implement BW, CSA, MoveToFront?
- What is the order of growth to form circular suffixes of a given string?
- What is wrong with using LSD.sort() in the program?
- Could we have used quicksort to sort suffixes? How? If so how can you avoid quadratic performance?

Burrows-Wheeler ctd…

- What are the 3-steps to burrows-wheeler transform?
- How would you sort strings w/o forming them explicitly?
- Is it necessary to do move to front? If not, why did we do it?
- How did we do the inverse transform?

Analysis of Algorithms

```
public static int f2(int N, int R) {
   int x = 0;
   for (int i = 0; i < R; i++)
      x += f1(i);
   return x;
}
```

Assume f1(N) is of O(N)

```
public static int f5(int N, int R) {
   int x = 0;
   for (int i = 0; i < N; i++)
      for (int j = 1; j <= R; j += j)
         x += f1(j);
   return x;
}
```

Analysis of Algorithms ctd..

Assume f1 is of O(N)

```
public static int f4(int N) {
   if (N == 0) return 0;
   return f4(N/2) + f1(N) + f1(N) + f1(N) + f4(N/2);
}
```

```
public static int f3(int N) {
   if (N == 0) return 1;
   int x = 0;
   for (int i = 0; i < N; i++)
      x += f3(N-1);
   return x;
}
```

Classifying Algorithms

(a) Which of the following can be performed in *linear time* in the *worst case*? Write P (possible), I (impossible), or U (unknown).

___ Printing the keys in a binary search tree in ascending order.

___ Finding a minimum spanning tree in a weighted graph.

___ Finding all vertices reachable from a given source vertex in a graph.

___ Checking whether a digraph has a directed cycle.

___ Building the Knuth-Morris-Pratt DFA for a given string.

___ Sorting an array of strings, accessing the data solely via calls to `charAt()`.

___ Sorting an array of strings, accessing the data solely via calls to `compareTo()`.

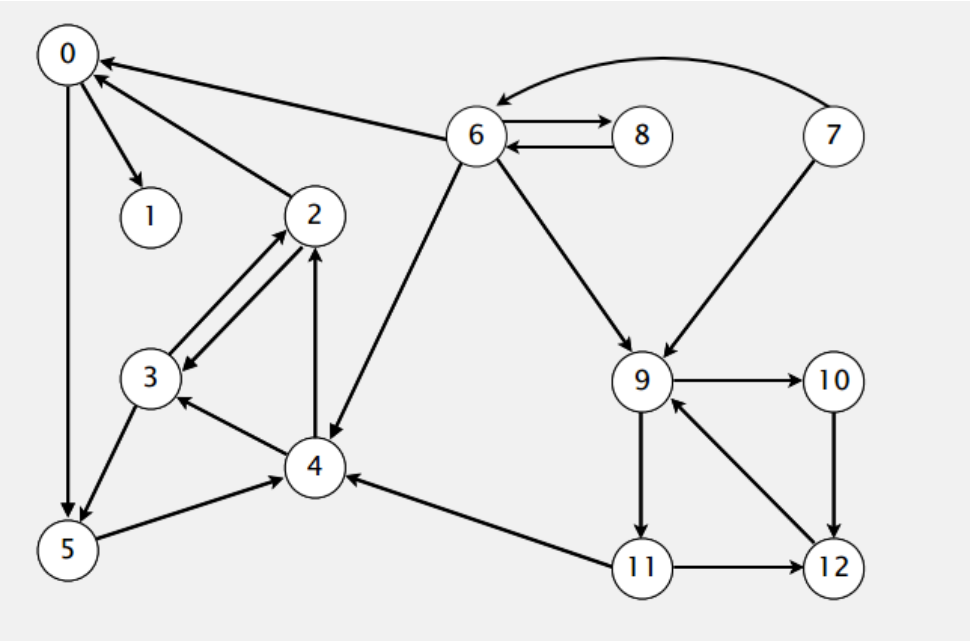
___ Finding the closest pair of points among a set of points in the plane, accessing the data solely via calls to `distanceTo()`.

Mystery Code

```
for (Edge e : G.adj(v))
{
   int w = e.to();
   if (dist[w] > dist[v] + e.weight())
   {
      dist[w] = dist[v] + e.weight();
      pred[w] = e;
      pq.insert(dist[w], w);
   }
}
```

This code belongs to one of the graph algorithms. Which one(s)?

Finding SCC's



What is the difference between **connected components** and **strongly connected components**?

Hashing

- When implementing a ST with hashing, what operations are not allowed in the ST?
- What is a collision and how do we avoid them?
- How can we minimize collisions in a hashtable?

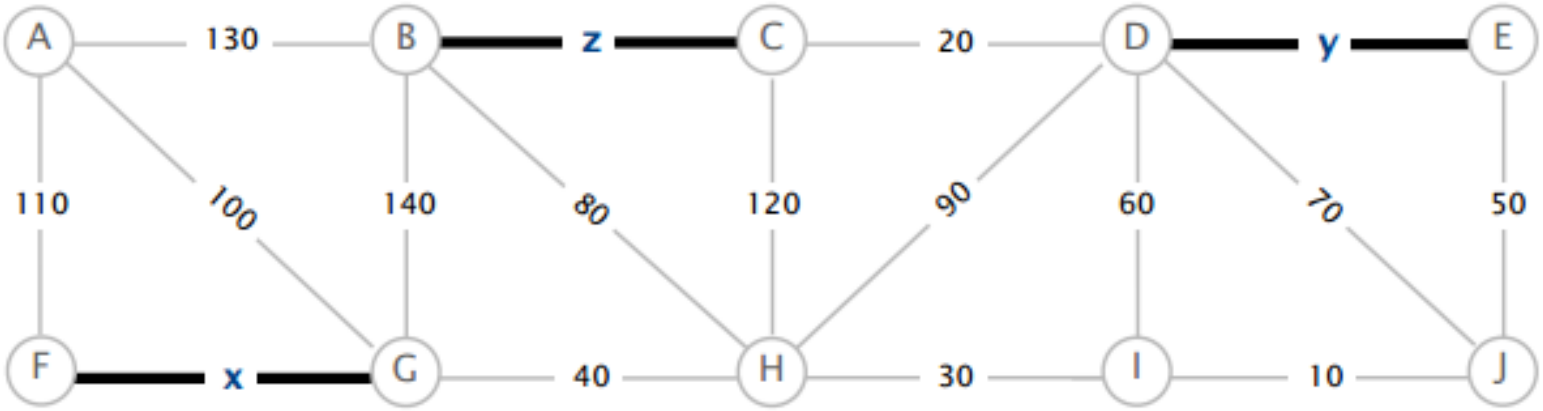
Hashing

- True/False
  - A linear probing hash table always finds a place
  - A quadratic probing hash table always find a place
  - A separate chaining hash table always finds a place
  - The load factor of a hash table is always <= 1
  - A linear probing hash table must be rehashed if load factor is over 0.7
  - A rehashed entry will be at the same location as the original

Hashing question

- Suppose 10,000 strings of length 5 from ASCII table is hashed using the hash function
  - $H(S) = \sum s[i]$ $I = 0 \ldots 4$ (table size = 10,000)
- Questions
  - Is this a good hash function? Why or why not?
  - If linear probing hash table is used, what is the probability of a collision?
  - If separate chaining is used, what is the average length of a chain?

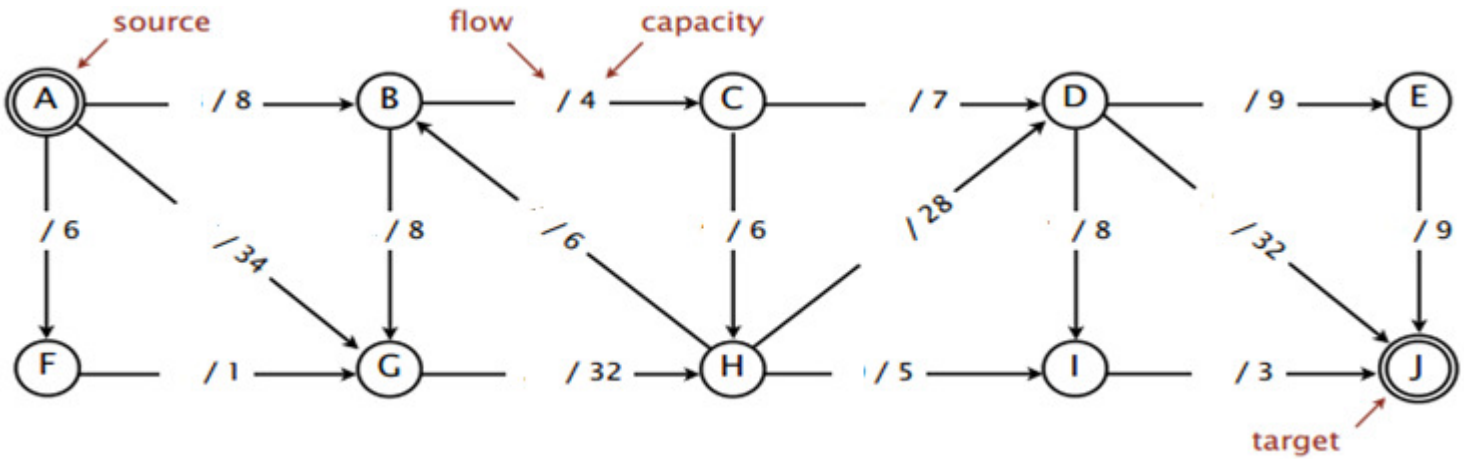
MST



If edges X, Y and Z are in the MST

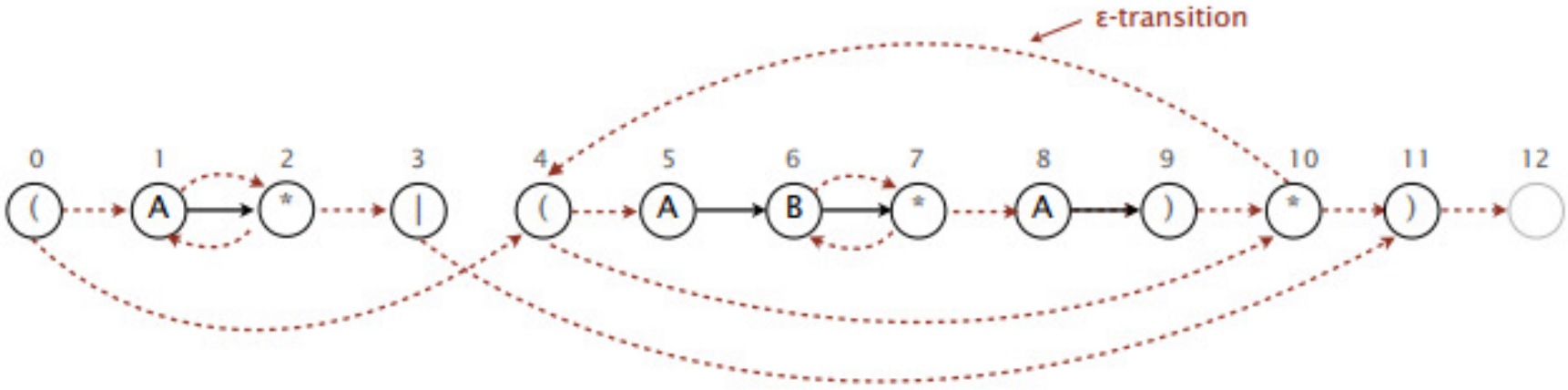
1. Find the other edges that are in MST
2. find upper bounds for edge costs of X, Y and Z?

Ford-Fulkerson



1. What is the possible max flow of the network?
2. Mark an augmenting path and increase flow
3. Find all augmenting paths and increase flow
4. What is the actual max flow?
5. What is a min-cut?
6. Min-cut can only be calculated when a certain condition is true. What is it? How do we find out?

Regular Expressions and NFA



1. What is the regular expression?
2. Suppose that you simulate the following sequence of characters on the NFA above: **A A A A A A A A** . In which one or more states could the NFA be?
3. Suppose that you want to construct an NFA for the regular expression (A * | (A B * A) +) where the operator + means one or more copies. What minimal change(s) would you make to the NFA above?

Compression Algorithms
Run-length encoding

- Describe the algorithm
- Under what circumstances would you use this algorithm?
- If 8-bit words are used to store counts, what is the length of the maximum run that can be stored?
- What can we do if the length of the run cannot be accommodated by n-bit word?
- What is the best case input for run-length encoding (8-bit code words)

Compression Algorithms
Huffman encoding

- Describe the algorithm
- What is the preprocessing step of the algorithm?
- What data structure(s) is/are used in the preprocessing step?
- If you compress a file with all characters the same (eg: 10000 A's) what is the compression ratio?
- Describe a situation where no compression is obtained

Compression Algorithms
LZW encoding

- Describe the algorithm
- What data structure is used to store code words?
- Is it possible to run out of code words to store new words? How?
- Should we send the code words with the compressed file?
- How can we decompress a file?
- What is the tricky case and how do we overcome that?

LZW

97
98
128
129
131
132
130

Decode the message a=97, b=98, and start next token from 128

KMP Algorithm

- Briefly describe the algorithm
- What is the order of growth of building the DFA? Typical algorithm? Best algorithm?
- What is the order of growth of the algorithm for searching for a pattern of length m in a text of length n?
- Can KMP be adjusted to find all occurrences of a pattern in a text? What is the order of growth?
- Is KMP the algorithm of choice for any substring search application?

Exercise

- Build the DFA for : ABAAABB

Exercise

	0	1	2	3	4	5	6	7	8	9	10
A	0	0								10	11
B					5		2				4
s										A	

Complete the table and find the search string

Boyer Moore (BM)

- Briefly describe the algorithm
- What is the pre-processing step of Boyer-Moore?
- What is the order of growth of the algorithm for searching for a pattern of length m in a text of length n? Best case? Worst case?
- Can BW be adjusted to find all occurrences of a pattern in a text? What is the order of growth?

Rabin-Karp (RK)

- Briefly describe the algorithm
- What is the pre-processing step of RK algorithm?
- What is the order of growth of the algorithm for searching for a pattern of length m in a text of length n? Best case? Worst case?
- Can RK be adjusted to find all occurrences of a pattern in a text? What is the order of growth?

Design Problem
connecting dorm rooms

There are N dorm rooms, each of which needs a secure internet connection. It costs $w_i > 0$ dollars to install a secure router in dorm room i and it costs $c_{ij} > 0$ dollars to build a secure fiber connection between rooms i and j. A dorm room receives a secure internet connection if either there is a router installed there or there is some path of fiber connections between the dorm room and a dorm room with an installed router. The goal is to determine in which dorm rooms to install the secure routers and which pairs of dorm rooms to connect with fiber so as to minimize the total cost.