

# 3/20 Exam 2 Review Arup

Tuesday, March 20, 2018 12:32 PM



Exam2

## Computer Science II Exam #2

Date: 3/26/2013

Last Name: Meade, First Name: Travis

1) (10 pts) You are attempting to create teams for the upcoming hackathon. Each team will consist of 2 students and you know that teams that have a total talent level of 100 or higher will advance to the next level. Given the talent level of each student at your school, devise an algorithm to maximize the number of pairs you choose with a talent level of 100 or higher. What is the run time of your algorithm on an input list of size  $n$ ? Execute your algorithm on a set of students whose talent levels are listed below:

86, 34, 99, 14, 55, 39, 32, 27, 61, 72, 8, 19, 81, 27, 76, 23, 64, 35, 45, 56

~~2~~, ~~5~~, ~~14~~, (19), ~~25~~, (27), (31), ~~32~~, ~~35~~, (39)<sup>\*</sup> →  
 (~~42~~), (~~53~~), (56), (~~61~~)<sup>\*</sup>, (64), (72), (76), (81), (86),  
 (99)  
 ↑

$ans = 0;$   
 $backpointer = n - 1;$   
 $sort(vals)$

```

for (int i = 0; i < backpointer; i++) {
  if (vals[i] + vals[backpointer] >= 100) {
    ans++;
    backpointer--;
  }
}
return ans;
  
```

List of Pairs: ( 2, 99 ), ( 14, 86 ), ( 19, 81 ), ( 27, 76 ), ( 31, 72 ), ( 39, 64 ),  
 ( 42, 61 ), ( 55, 56 ), (     ,      ), (     ,      ), (     ,      ), (     ,      ),

(Note: More slots are provided than necessary.)

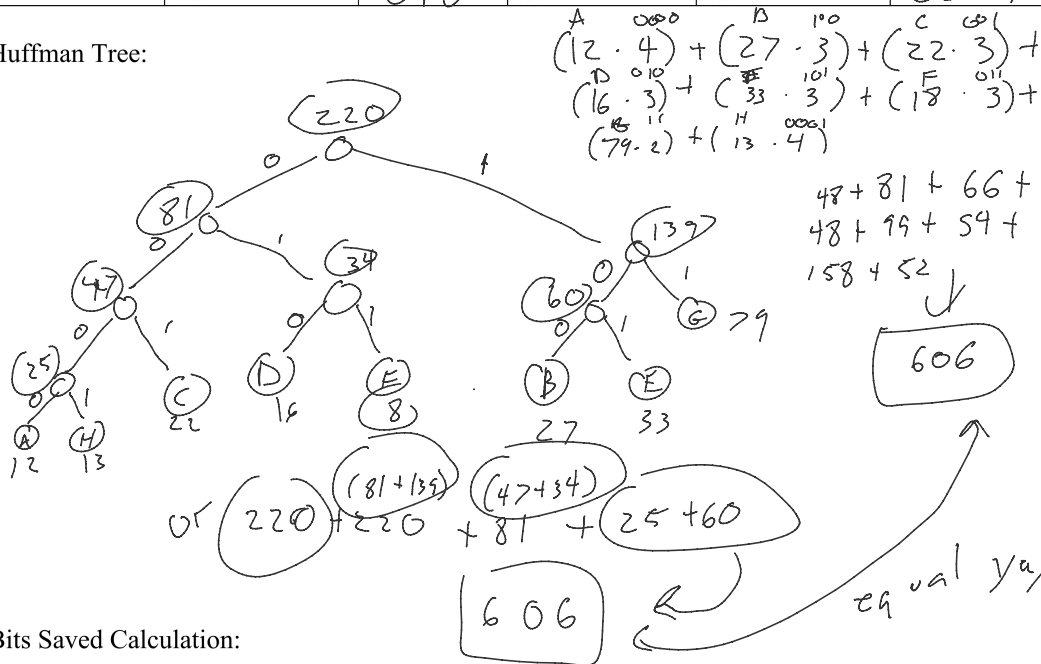
Maximum Number of Pairs = 8

2) (15 pts) Consider creating a Huffman code for a file with eight distinct characters, 'A' through 'H'. Given the character frequencies below, create a Huffman tree, determine valid Huffman codes and calculate the number of bits saved (not counting the storage of the code) by the code if the old fixed-length storage scheme used 3-bits per character.

*Need to sort Freq least to greatest*

| Character | Frequency | Code | Character | Frequency | Code |
|-----------|-----------|------|-----------|-----------|------|
| A         | 12        | 0000 | E         | 33        | 101  |
| B         | 27        | 100  | F         | 18        | 011  |
| C         | 22        | 001  | G         | 79        | 11   |
| D         | 16        | 010  | H         | 13        | 0001 |

Huffman Tree:



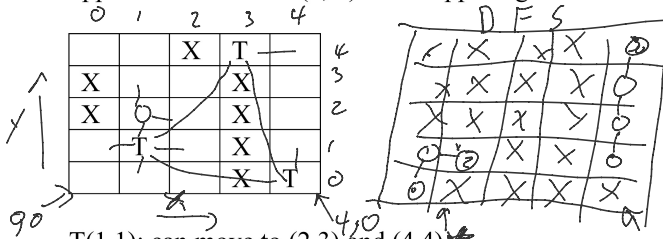
Bits Saved Calculation:

$$660 - 606 = 54 \text{ bits saved}$$

↑ orig      ↑ after

For questions 3 and 4, you will perform a depth first search and breadth first search on the modified grid below, starting at square (0,0). Squares containing an X can not be visited. Squares containing a T are teleport squares. Each teleport square has two possible neighbors to visit. These will be explicitly listed below. From all other squares (x,y), you may visit the neighbors (x-1, y), (x, y-1), (x+1, y) and (x, y+1), so long as those square can be visited. In the case of choosing between neighbors in either algorithm, always choose the one with the minimum x coordinate first. If there are multiple such neighbors, choose the one with the minimum y coordinate first.

Here is the grid, with (0,0) in the lower left corner, (4,0) in the lower right corner, (0, 4) in the upper left corner and (4, 4) in the upper right corner.



T(1,1): can move to (2,3) and (4,4)  
 T(4,0): can move to (0,4) and (2,2)  
 T(3,4): can move to (0,1) and (4,0)

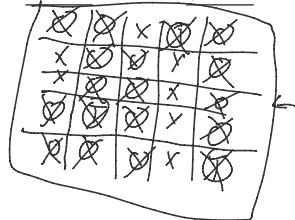
min X First  
 min Y Second

3) (9 pts) Show the order that the grid squares get visited, starting from (0,0), in a depth first search, using the rules above:

1. (0,0)
2. (0,1)
3. (1,1)
4. (2,3)
5. (1,3)
6. (1,2)
7. (2,2)
8. (2,1)
9. (2,0)
10. (1,0)
11. (1,4)
12. (0,4)
13. (4,4)
14. (3,4)
15. (4,0)
16. (4,3)
17. (4,2)
18. (4,1)

4) (9 pts) Show the order that the grid squares get visited, starting from (0,0), in a breadth first search, using the rules above:

- 1. (0,0)
- 2. (0,1)
- 3. (1,0)
- 4. (1,1)
- 5. (2,0)
- 6. (4,3)
- 7. (4,4)
- 8. (2,1)
- 9. (1,3)
- 10. (2,2)
- 11. (3,4)
- 12. (4,3)
- 13. (1,2)
- 14. (1,4)
- 15. (4,0)
- 16. (4,2)
- 17. (0,4)
- 18. (4,1)





5) (10 pts) Consider the following topics that may be taught in Computer Science II:

- a) Divide and Conquer Introduction
- b) Kruskal's Algorithm
- c) DP – 0-1 Knapsack Algorithm
- d) Disjoint Sets
- e) Subset Sum Problem
- f) Topological Sort
- g) Greedy Algorithm Introduction
- h) Depth First Search
- i) Dynamic Programming Introduction
- j) Skyline Algorithm

The dependencies in teaching these topics are listed below. (Note: an ordered pair (x, y) in the list means that topic x must be taught before topic y.)

(a, e), (a, h), (a, i), (a, j), (d, b), (e, c), (g, b), (h, f), and (i, c).

Run a Topological Sort to provide a valid ordering of teaching these topics. In particular, **use the algorithm based on DFS shown in class, which fills in the list of items from the back.** Make sure to make the initial DFS calls in alphabetical order, as necessary. In any depth first search, when given a choice, choose the letter that comes first alphabetically. Provide the ordering below. (Note: Your grade will solely be based on your answer, with each slot being worth 1 point.) Please just write out the letter for each topic instead of the full topic itself. You must clearly write each letter to get credit.

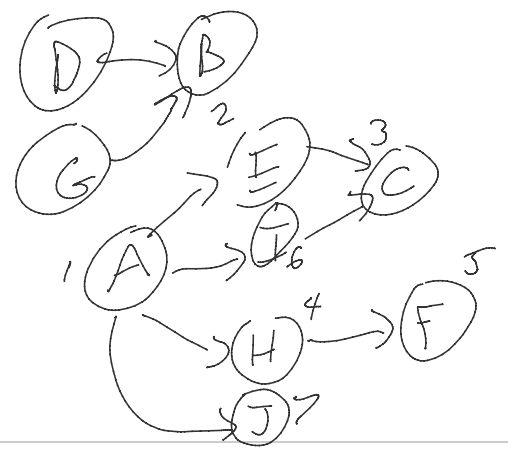
- |             |              |
|-------------|--------------|
| 1. <u>G</u> | 6. <u>I</u>  |
| 2. <u>D</u> | 7. <u>H</u>  |
| 3. <u>B</u> | 8. <u>F</u>  |
| 4. <u>A</u> | 9. <u>E</u>  |
| 5. <u>J</u> | 10. <u>C</u> |

Alpha from start reverse at (A)

A on stk  
 E on stk  
 C on stk  
 pop C as 10  
 pop E as 9  
 H on stk  
 F on stk  
 pop F as 8  
 pop H as 7

I on stk  
 pop I as 6  
 J on stk  
 pop J as 5  
 pop A as 4  
 D on stk  
 B on stk  
 pop B as 3

pop D as 2  
 G on stk  
 pop G as 1





7) (20 pts) Arup's friend Jeff really loves Four-Loko. Unfortunately, due to the dangerous nature of the drink, many places are looking to ban it. Jeff has decided to hatch a plan so that he and his friends can procure as much Four-Loko as possible, just in case it becomes illegal. You are given a list of people,  $p_1$  through  $p_n$ , each who desire to stock up on Four Loko. Each of them has a limit,  $L_i$  ( $1 \leq i \leq n$ ), for how many cases they can afford. Furthermore, there are  $f$  possible flavors of cases they can obtain. For each of the  $f$  possible flavors, each of his friends has a maximum limit  $M_{i,j}$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq f$ ) they are willing to buy. Finally, there are  $S$  stores, each of which stocks Four Loko. Let  $S_{i,j}$  denote the number of cases of flavor  $j$  ( $1 \leq j \leq f$ ) that store  $i$  ( $1 \leq i \leq S$ ) has in stock. We wish to calculate the maximum number of cases that Jeff and his friends can collectively procure under these constraints. Describe an algorithm to solve this problem using a flow network. Justify informally why your algorithm will produce the correct answer. (Note: Be as specific and unambiguous as possible in describing your algorithm.)

8) (12 pts) The adjacency matrix below stores the capacities for a flow network. Answer the questions that follow the adjacency matrix. All vertices that are not connected by an edge are denoted by a 0.

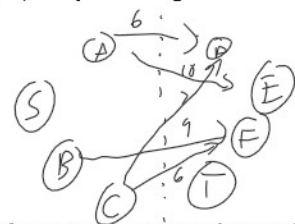
|   | S | A  | B | C  | D | E  | F | T  |
|---|---|----|---|----|---|----|---|----|
| S | 0 | 15 | 5 | 10 | 0 | 0  | 0 | 0  |
| A | 0 | 0  | 0 | 0  | 6 | 10 | 0 | 0  |
| B | 0 | 0  | 0 | 0  | 0 | 0  | 9 | 0  |
| C | 0 | 0  | 0 | 0  | 7 | 0  | 6 | 0  |
| D | 0 | 0  | 0 | 0  | 0 | 0  | 0 | 15 |
| E | 0 | 0  | 5 | 0  | 0 | 0  | 0 | 4  |
| F | 0 | 0  | 0 | 0  | 0 | 0  | 0 | 8  |
| T | 0 | 0  | 0 | 0  | 0 | 0  | 0 | 0  |

6+10+7+6  
+7

a) What vertex is the source of this flow network? S

b) What vertex is the sink of this flow network? T

c) Calculate the value of the cut  $\{S, A, B, C\}$  and  $\{D, E, F, T\}$  only with regard to the capacities. (Hint: just add the capacities of all the forward edges.)



$X \notin S' \Rightarrow$   
 $X \in T'$   
L.T  
 $S' = \{S, B, D, F\}$   
 $T' = \{A, C, E, T\}$   
as an exercise

38

d) Determine the maximum flow of this network. Please show each augmenting path that you add and the order that you add each path in the chart below.

| Added Path (list each vertex in the path) | Flow Added(value) |
|---|-------------------|
|   |                   |
|   |                   |
|   |                   |
|   |                   |
|   |                   |
|   |                   |
|   |                   |
|   |                   |

Total Flow = \_\_\_\_\_

9) (3 pts) Which NYC Mayor is part of the group that founded Bloomberg L. P.? \_\_\_\_\_

**Scratch Page – Please clearly mark the work on this page you would like graded.**