Kamryn Parker

CS 321 Data Structures (Spring 2020)

Homework #3 (85 points), Due at 11:00PM on 4/30/20 (Thursday)

- **Q1(11 points): Running time, and asymptotic notations**

  (a)(3 pts)(Multiple choice) If an algorithm's running time can be expressed as a function $f(n) = O(n^2) + n \log n$, then which one of the following asymptotic bounds is possible for the running time?

  1. $\Theta(\log n)$ The largest O in the equation is n^2
  2. $\Theta(n)$
  3. $\Theta(n^2)$
  4. $\Theta(n^3)$

  (b)(3 pts) Assume an algorithm runs in $\Theta(n^2)$, then which one of the following is a correct asymptotic notation for it?

  1. $O(n^3)$ ⟵ possible but unsure
  2. $\Omega(n^2)$ ⟵ Also possible
  3. $\Omega(n)$ ⟵ Also possible
  4. All the above are correct

  (c)(5 points) Please re-arrange (sort) the following functions based on their growth rates, from the least to the greatest. In addition, please underline those functions which have the same asymptotic bound.

  $$\cancel{\sqrt{n}},\ \cancel{3^{n/2}},\ \cancel{\log_e n},\ \cancel{n\log_{10} n},\ \cancel{n!},\ \cancel{2^n},\ \cancel{\log_2(n!)},\ \cancel{2^{1000}},\ \cancel{(\log_2 n)^n}$$

  2^{1000}, $\log_e n$, $\sqrt{n}$ ($\downarrow$ $n^{1/2}$), $\underline{n \log_{10} n,\ \log_2(n!)}$, $3^{n/2}$, 2^n, $(\log_2 n)^n$, $n!$

  constants → logs not multiplied, n, n*log n, exponentials, n!

- **Q2(19 points): Basic Data Structures**

  (a)(3 pts)(Multiple choice) Which one of the following is the true statement for linked lists (assume there is only one entry point to the list - `Head`)?

  1. It takes linear time to insert an element to the front of a singly non-circular list.
  2. It takes linear time to insert an element to the end of a singly circular list.
  3. It takes linear time to insert an element to the end of a doubly non-circular list.
  4. It takes linear time to insert an element to the end of a doubly circular list.

Top

(b)(3 pts)(Multiple choice) If we read a sequence of items $< A, B, C, D, E, F >$ from a file and maintain some of the items in a stack by `push` and `pop` operations, which one of the following stack is possible? (Assume that each item can be pushed into the stack only once).

1. top $< D, E, F >$ bottom
2. top $< F, C, A >$ bottom
3. top $< C, D, E >$ bottom
4. top $< E, D, F >$ bottom

Bottom

F ↓ E ↓ D ↓ C ↓ B ↓ A

(c)(3 pts)(Multiple choice) For hashing by open addressing, we need to decide the table size m and some parameters so that the table can be fully utilized. What does that actually mean?

1. Allocate a table with size m equal to the number of elements stored.
2. Decide m and other parameters so that for any element, the initial m probe positions are distinct.
3. Decide m and other parameters to reduce the probability of collisions.
4. Decide m and other parameters so that no collision could happen.

(d)(10 pts) Suppose we would like to insert a sequence of numbers $< 31, 72, 55, 7, 39 >$ into a hash table with table size 8 using the three open addressing methods, with the primary hash function $h_1(k) = k \bmod 8$, the secondary hash function

$$h_2(k) = \begin{cases} 1 + (k \bmod 7) & \text{if } k \bmod 7 \text{ is even} \\ k \bmod 7 & \text{if } k \bmod 7 \text{ is odd} \end{cases}$$

and the constants $c_1 = c_2 = 1/2$ (in quadratic probing). Please insert numbers into the tables below.

index	linear	quadratic	double
0	72	72	72
1	55	39	
2	7	55	
3	39		39
4			
5		7	55
6			
7	31	31	31

$[(55 \bmod 8) + \frac{1}{2}(1) + \frac{1}{2}(1)^2] \bmod 8 =$

$7 + 1 \bmod 8 = 0$

$h(55) [7 + \frac{1}{2}(2) + \frac{1}{2}(2)^2] \bmod 8 = 2$

$h(7) \quad 7 + \frac{1}{2}(3) + \frac{1}{2}(3)^2 = 13 \bmod 8 =$

$h(39) \quad 7 + \frac{1}{2}(4) + \frac{1}{2}(4)^2 = 17 \bmod 8$

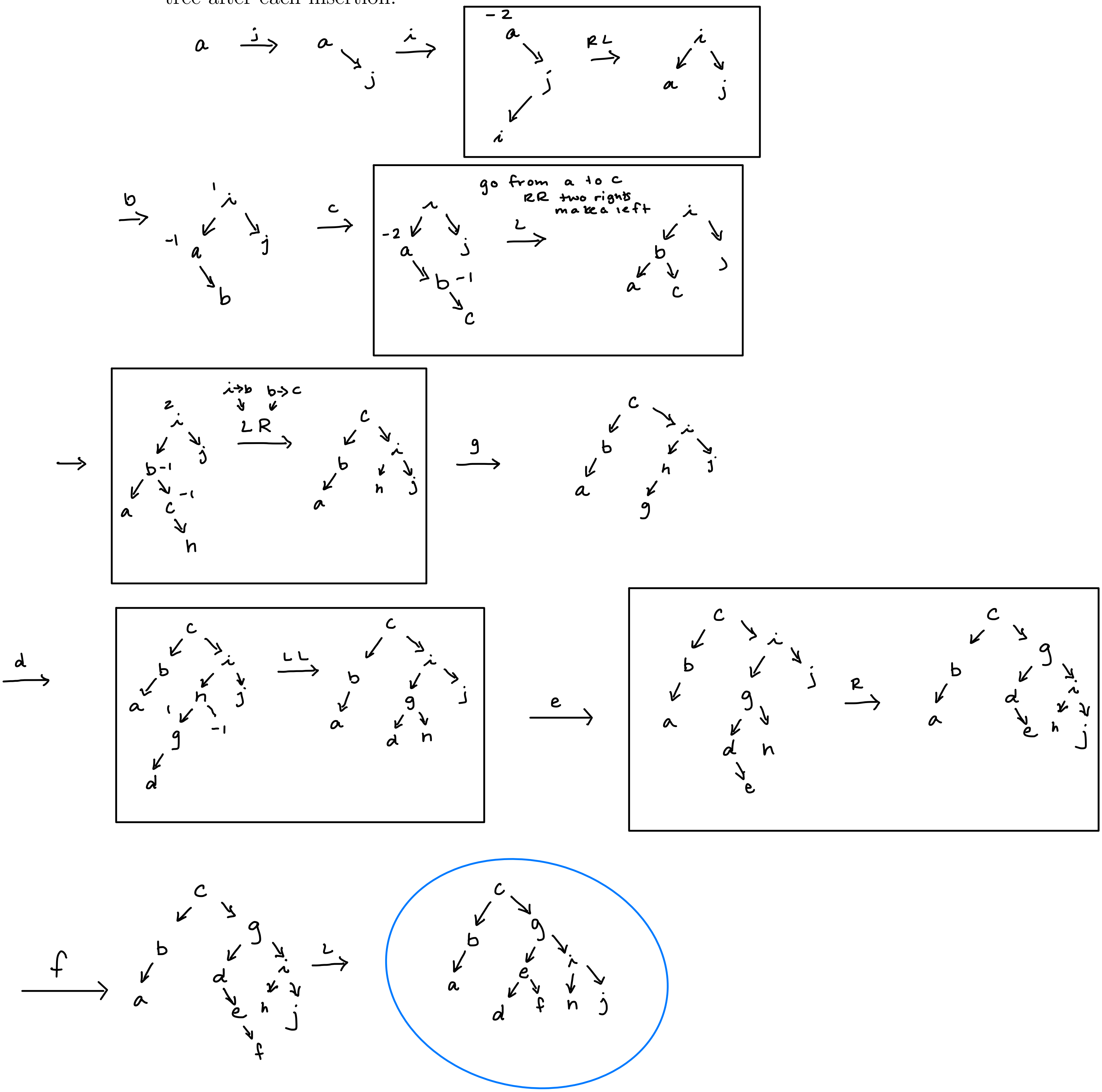
$h(55) = [7 + 1(55 \bmod 7)] \bmod 8 = 13 \bmod 8 = 5$

$h(7) = [7 + 1(7 \bmod 7)] \bmod 8 = 7$

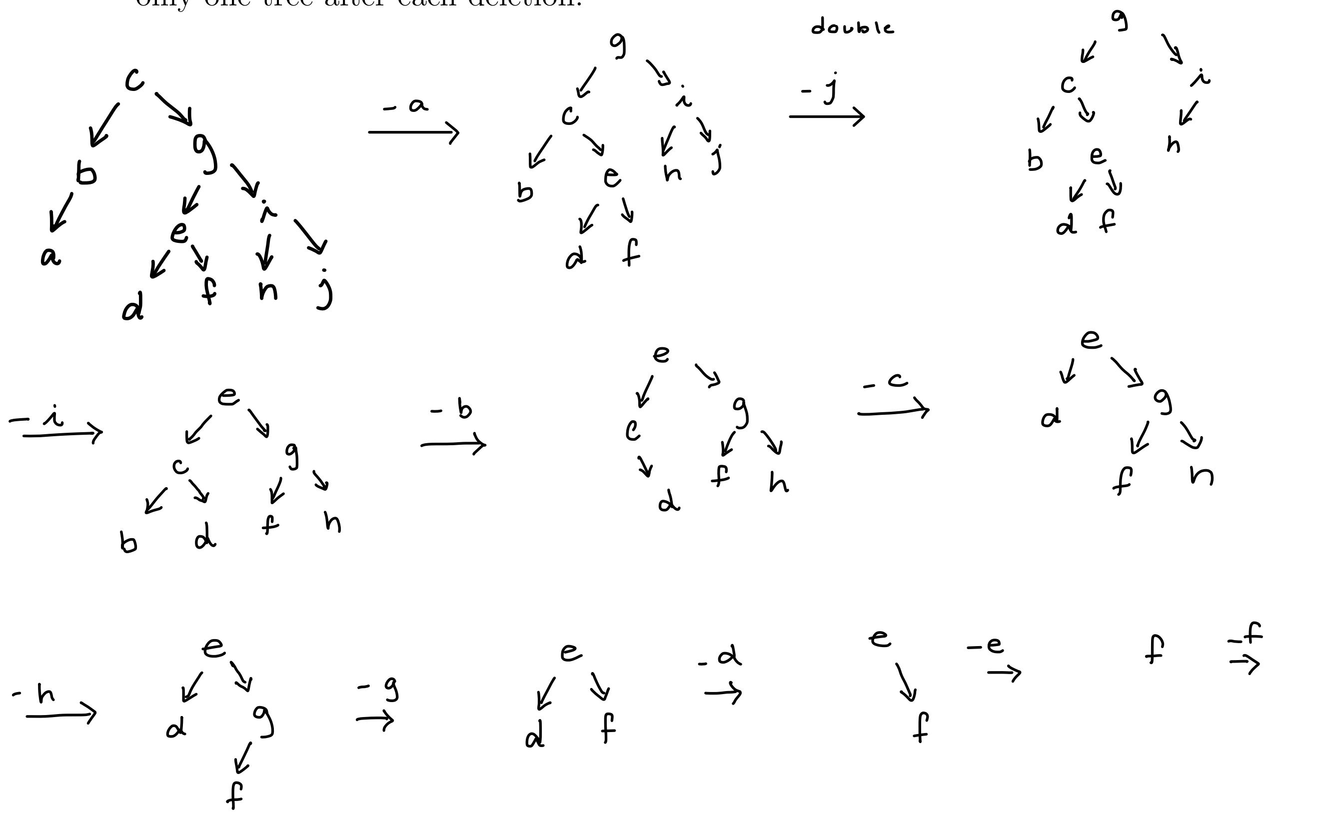
$h(39) = [7 + 1(39 \bmod 7)]$

- **Q3(20 points): AVL Trees**

  (a)(10 pts) What are the sequence of AVL tress after inserting each character in the list $< a, j, i, b, c, h, g, d, e, f >$ to an initially empty AVL tree? Note: please draw only one tree after each insertion.



(b)(10 pts) From the AVL tree you have built in part (b), what are the sequence of trees after deleting each character in the list (i.e., delete a , delete j, ...)? Note: please draw only one tree after each deletion.

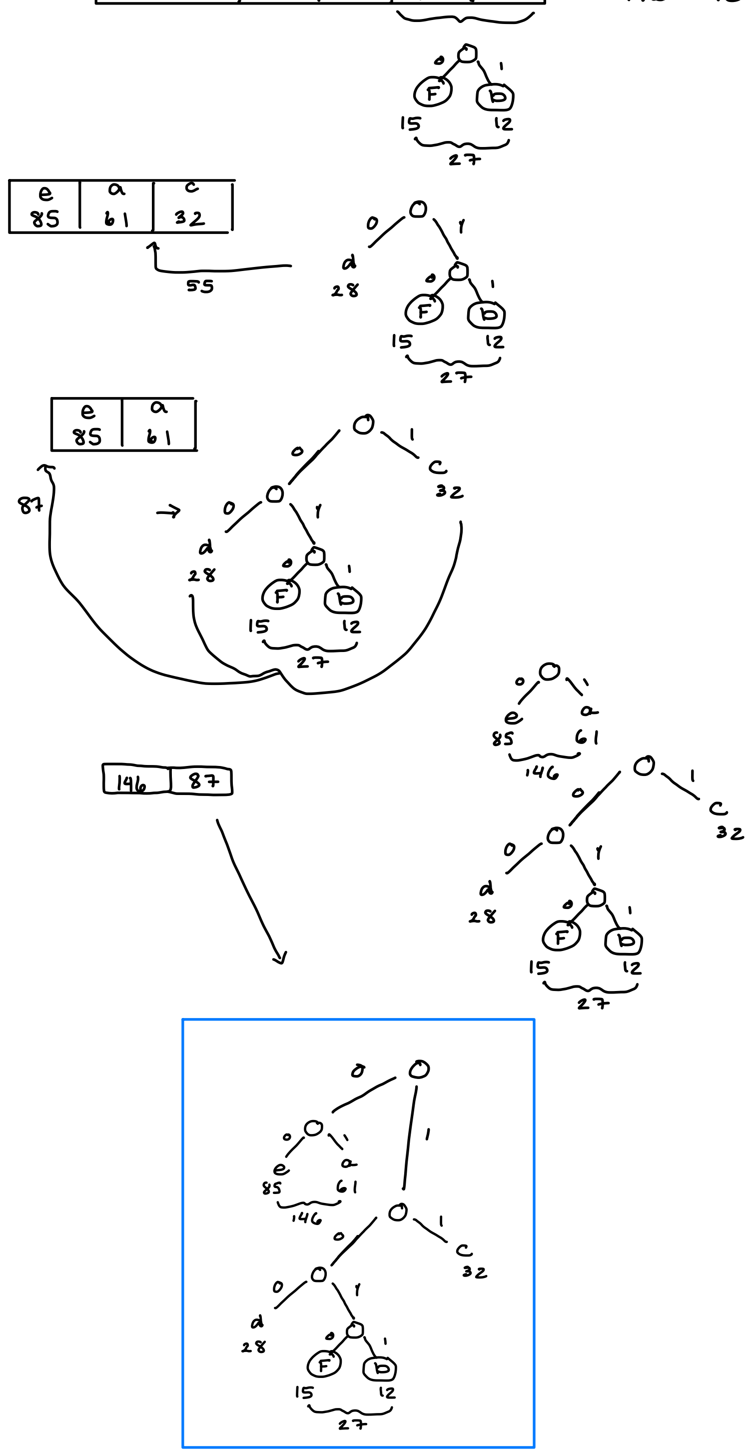


- **Q4(10 points): Huffman Trees**

  Given a text file with only six different characters $\{a, b, c, d, e, f\}$, the frequencies of these characters in the file are $\{(a : 61), (b : 12), (c : 32), (d : 28), (e : 85), (f : 15)\}$. Based on these frequencies, please construct a Huffman tree with only six leaf nodes (one for each of these six characters). Please show the intermediate forests on each step.

e	a	c	d	f	b
85	61	32	28	15	12

heaviest frequency goes to the left



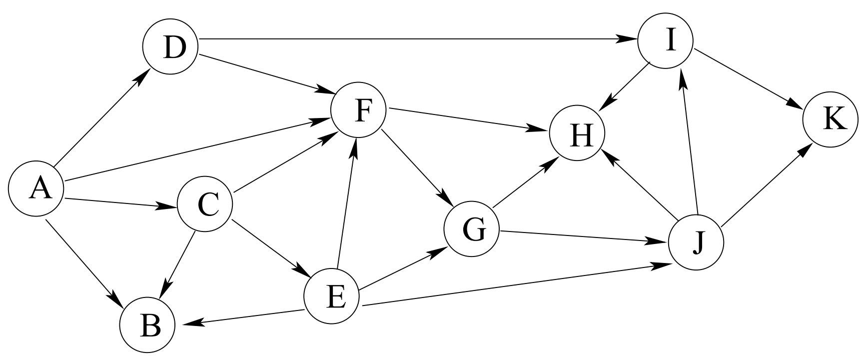
- **Q5(7 points): Tree Traversal**
  Given a binary tree, please write a pseudocode to perform a level-order traversal to print all nodes in the tree.

```
Level-order-print(x)    // x is the root of the tree    go from element to its sibling
{

  Queue q ;

  while (!q .isEmpty() ) {
    x = dequeue();
    print ( x .data) ;
    enqueue ( x .left) ;
    enqueue ( x .right);

      Start from beginning then enqueue everything down. Children are enqueued
      after the previous level has been queued.

}
```

- **Q6(18 points): Graph Algorithms**

  A weighted and directed graph is given below.

use notes



(a)(4 pts) Based on the DAG above, please find the discovering sequence of vertices in a BFS search if A is the source vertex.

A, BCDF, EGHI, JK

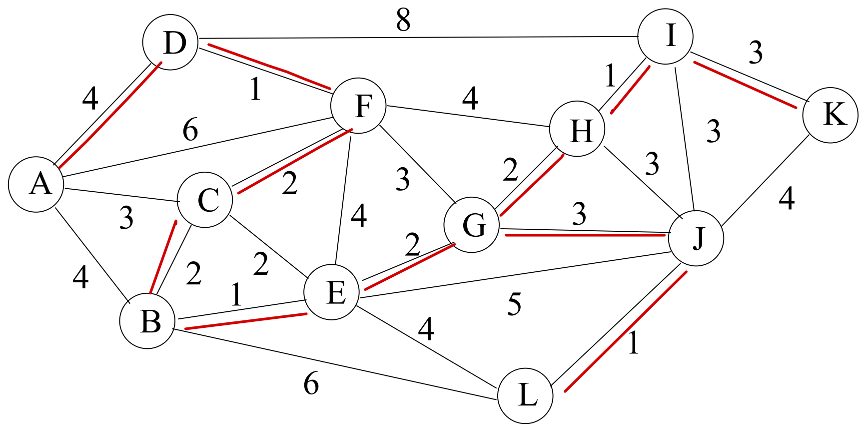
(b)(4 pts) Based on the DAG above, please find the discovering sequence of vertices in the DFS search, assuming that during the search if there are multiple vertices can be discovered next, please discover vertices based on their alphabetical order.

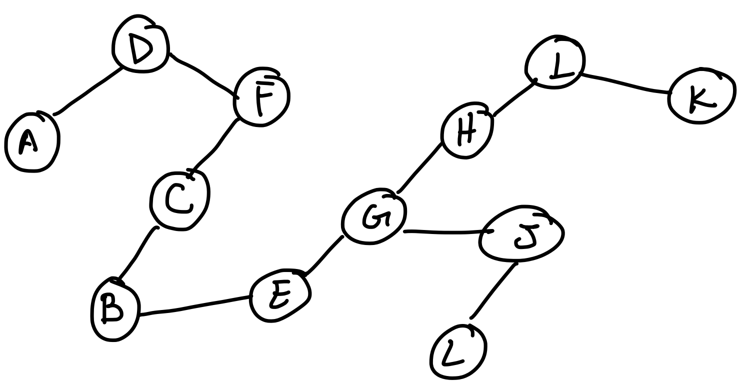
A, B, C, E, F, G, H, J, I, K, D

(c)(4 pts) Based on the DAG above, please find the topological sequence of vertices, assuming that during the DFS search if there are multiple vertices can be discovered next, please discover vertices based on their alphabetical order.

A, D, C, E, F, G, J, I, K, H, B

(d)(6 pts) Please find a (any) minimum spanning tree of the graph below





(D, F)	1	✓	(H, J)	3	x
(L, J)	1	✓	(I, J)	3	x
(H, I)	1	✓	(I, K)	3	✓
(B, E)	1	✓	(A, D)	4	x
(G, H)	2	✓	(A, B)	4	x
(G, E)	2	✓	(E, F)	4	x
(C, F)	2	✓	(F, H)	4	x
(C, B)	2	✓	(E, L)	4	x
(C, E)	2	x	(J, K)	4	x
(A, C)	3	✓	(E, J)	5	x
(F, G)	3	x	(A, F)	6	x
(G, J)	3	✓	(B, L)	6	x
			(D, I)	8	x