CSE 373 Spring 2001: Sample Final Exam Solutions

1. **Big Oh and Theta** (a and b: 5 and 5 points)
   a. Which of the following statements is/are true:
      i. $N^2 + N \log N$ is $\Theta(N^2)$
      ii. $N^2 - 17$ is $\Omega(N^2)$
      iii. $15N + \log N$ is $O(\log N)$
      iv. 2^N is $o(N^{100})$

   Answer: i and ii

   b. What is the running time T(N) of the following code fragment in Θ notation as a function of N? Explain your answer.

```
int result = 0;
int i = N;
while (i >= 1)
{
    result++;
    i = i / 2;
}
```

   Answer: $T(N) = \Theta(\log N)$. Here, $i = N/2^j$. Loop is executed k times, until: $N/2^k < 1$ which implies $k = (\log N)+1$, which implies $T(N) = \Theta(\log N)$.

2. **Recurrence Relations and Run Time Analysis** (a and b: 4 and 6 points)
   Consider the following recursive function:

```
int Lessby3(int N){
 if (N < 3) return 3;
 else return Lessby3(N-3)*Lessby3(N-3);
}
```

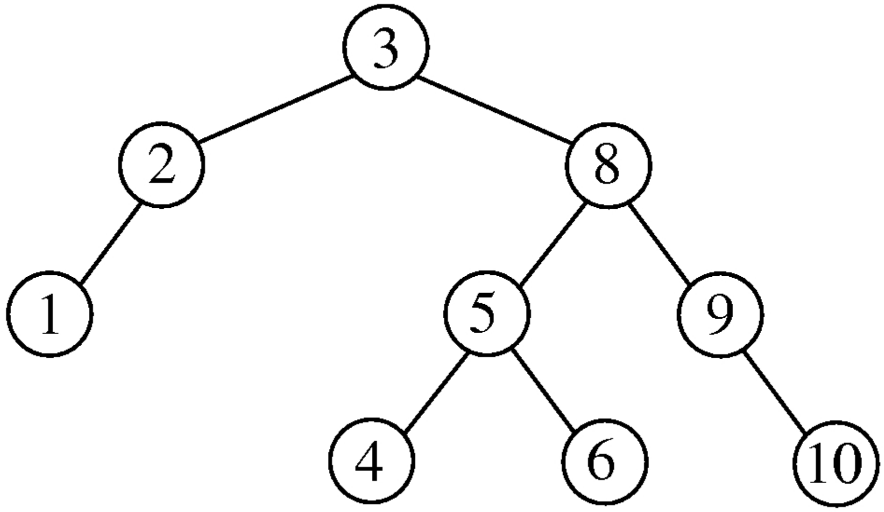
   a. Suppose T(N) is the running time of the above function for input N. Write down the recurrence relation for T(N).

   Answer: $T(N) = 2T(N-3) + c$ (for $N >= 3$ and c = constant)
   $T(N) = c_0$ for $N < 3$ (c_0 is another constant).

   b. Solve your recurrence relation in (a) to get an expression for T(N). What is the running time T(N) in Θ notation?

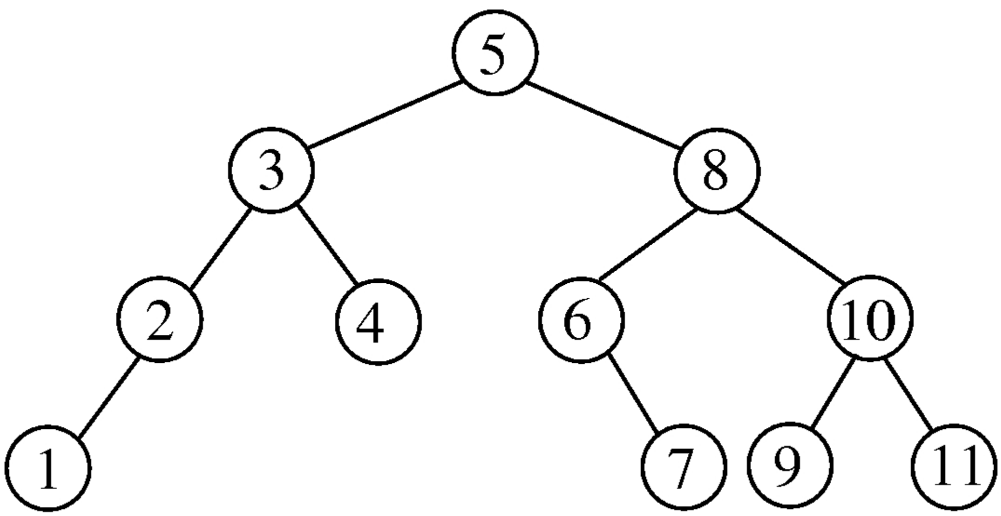
   Answer: $T(N) = 2T(N-3) + c = 2(2T(N-6) + c) + c = 2(2(2T(N-9) + c) + c) + c = \ldots$
   $= 2^i\, T(N-3i) + 2^{i-1} c + 2^{i-2} c + \ldots + c = 2^{N/3}\, T(0) + (2^{N/3} - 1)c = \underline{\Theta(2^{N/3})}$

3. **AVL and Splay Trees** (a and b: 5 and 5 points)
The following two questions are based on the following tree:



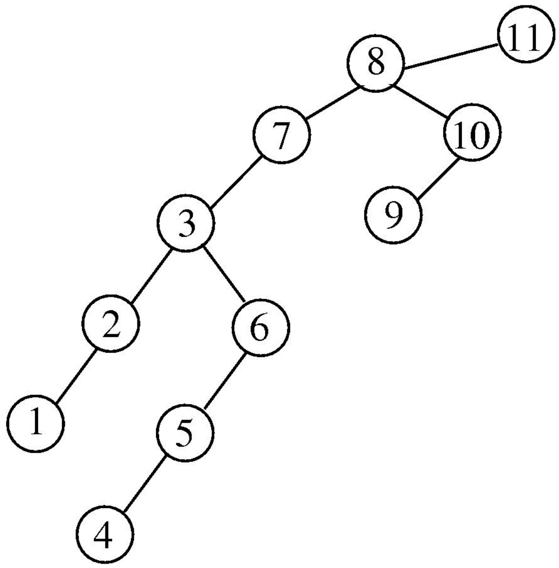
a. Suppose the above tree is an AVL tree. Draw the AVL tree that results from inserting 7 followed by 11 into the above AVL tree.

Answer:



b. Suppose the above tree is a splay tree. Draw the tree that results from inserting 7 followed by 11 into the above splay tree.

Answer:



4. **Priority Queues** (a, b, and c: 3, 2, and 5 points)

a. Draw the binary heap (a min-heap) that results from inserting the sequence of integers 10, 12, 1, 14, 6, 5 into an initially empty binary heap.

Answer:

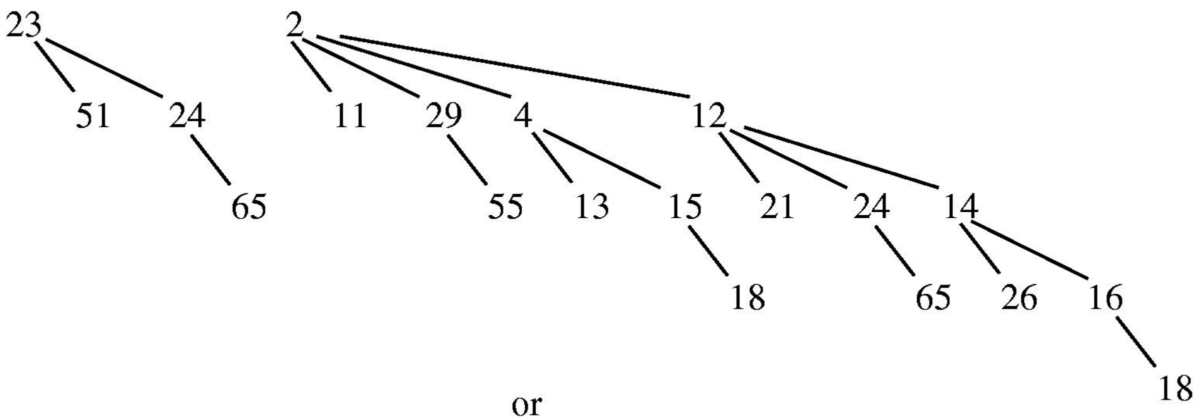
```
         1
        / \
      6     5
     / \   /
   14  12 10
```

b. Recall that the binomial tree of height k is formed by combining two binomial trees of height k-1, with height zero represented by a one-node tree. A binomial queue contains some combination of binomial trees. How many binomial trees can there be in a binomial queue with N nodes?

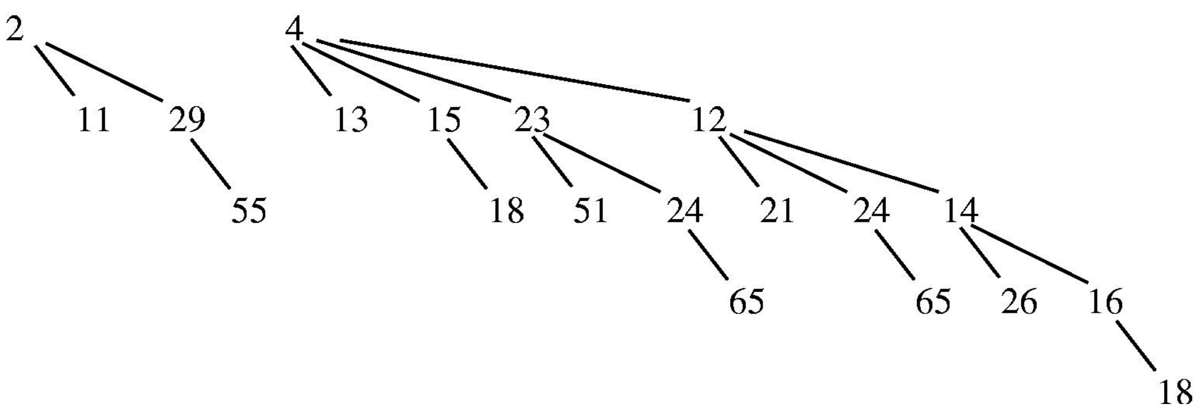
Answer: O(log N)

c. Let Q1 be the binomial queue that results from inserting the integers 12, 21, 24, 65, 14, 26, 16, 18, 23, 51, 24, 65, 13 (in that order) into an empty binomial queue. Let Q2 be the binomial queue that results from inserting the integers 2, 11, 29, 55, 15, 18, 4 (in that order) into an empty binomial queue. Draw the result of merging queues Q1 and Q2.

Answer:



or



5. **Hashing** (a, b, and c: 4, 4, 2 points)
Consider the hash function Hash(X) = X mod 9 and the ordered input sequence of keys 51, 23, 73, 99, 44, 79, 89, 38. Draw the result of inserting these keys in that order into a hash table of size 9 (cells indexed by 0, 1, …, 8) for the following two collision resolution strategies:

a. open addressing with linear probing, where F(i) = i

0	99
1	73
2	89
3	38
4	
5	23
6	51
7	79
8	44

b. open addressing with double hashing, where $hash_2(X) = 7 - (X \bmod 7)$ and $F(i) = i \cdot hash_2(X)$

0	99
1	73
2	38
3	89
4	
5	23
6	51
7	79
8	44

c. What is the load factor of the hash tables in (a) and (b)?

Answer: 8/9 = 0.89 for both (a) and (b)

6. **Simple Sorts and Heapsort** (a, b, and c: 4, 2, and 4 points)

a. Sort the array 34, 8, 64, 51, 32, 21 using Insertion Sort. Write your answer in the form of a table showing the array after each pass.

Answer: (Same as Figure 7.1 in textbook)

Original	**34**	**8**	**64**	**51**	**32**	**21**
After p =1	8	34	64	51	32	21
After p =2	8	34	64	51	32	21
After p =3	8	34	51	64	32	21
After p =4	8	32	34	51	64	21
After p =5	8	21	32	34	51	64

b. What is the running time of insertion sort for input size N if
   i. the input is already sorted?
   ii. the input is in reverse order?

   Choose the best upper bound from: O(log N), O(N), O(N log N), and $O(N^2)$

Answer: i. O(N)   ii. $O(N^2)$

c. Sort the sequence 7, 3, 1, 5, 4 using <u>Heapsort</u>. First show the result of BuildHeap and then the result after each DeleteMax. Draw both the tree-structured heap and the input array.

Answer: See the solution for Question 6c in Homework #4.

7. **Mergesort and Quicksort** (a and b: 6 and 4 points)

a. Sort the sequence 3, 1, 4, 1, 5, 9, 2, 6 using <u>Mergesort</u>. Show the sequence of elements after each of the recursive "divide-in-half" steps as well as the result of each of the merge steps.

Answer: Same as the solution for Question 6d in Homework #4.

b. Which of the following statements is/are true (write down all that you think are true – use the standard implementation of Mergesort and Quicksort as discussed in class and in the textbook):
   i. Mergesort is an "in-place" sorting algorithm
   ii. Quicksort has $O(N^2)$ worst case running time
   iii. Mergesort and Quicksort both have O(N log N) worst running time
   iv. Quicksort is an "in-place" sorting algorithm
   v. Quicksort is a "stable" sorting algorithm

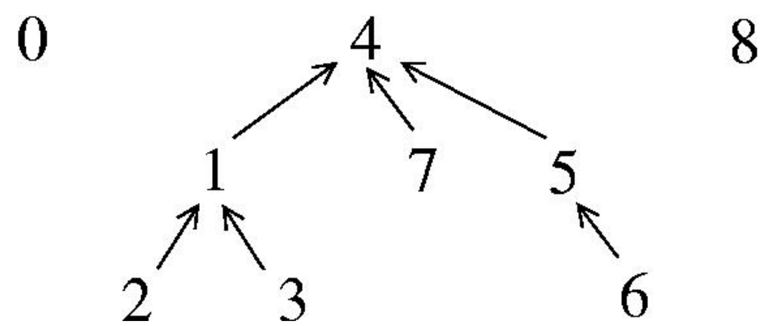
Answer: ii and iv

8. **Union-Find** (a and b: 5 and 5 points)

Consider the set of initially unrelated elements 0, 1, 2, 3, 4, 5, 6, 7, 8.

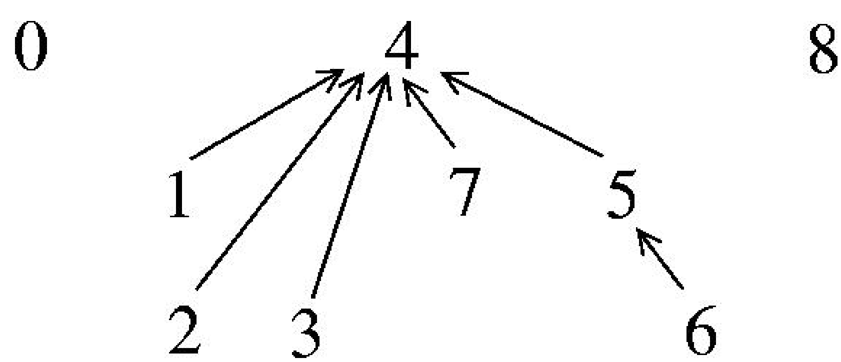
a. Recall the <u>union-by-size</u> operation. Draw the final forest of up-trees that results from the following sequence of operations based on union-by-size: Union(1,2), Union(3,1), Union(4,7), Union(5,6), Union(4,5), Union(1,4)

Answer:



b. Recall what path compression does. Draw the new forest of up-trees that results from doing a Find(2) with path compression, followed by a Find(3) with path compression on your forest of up-trees from (a).

Answer:



9. **Shortest Path and MST** (a and b: 6 and 4 points)
You can fill in this page and submit it with your other answer sheets

a. Fill in the following tables using Dijkstra's algorithm for single-source shortest path for the given directed graph, with source = C:

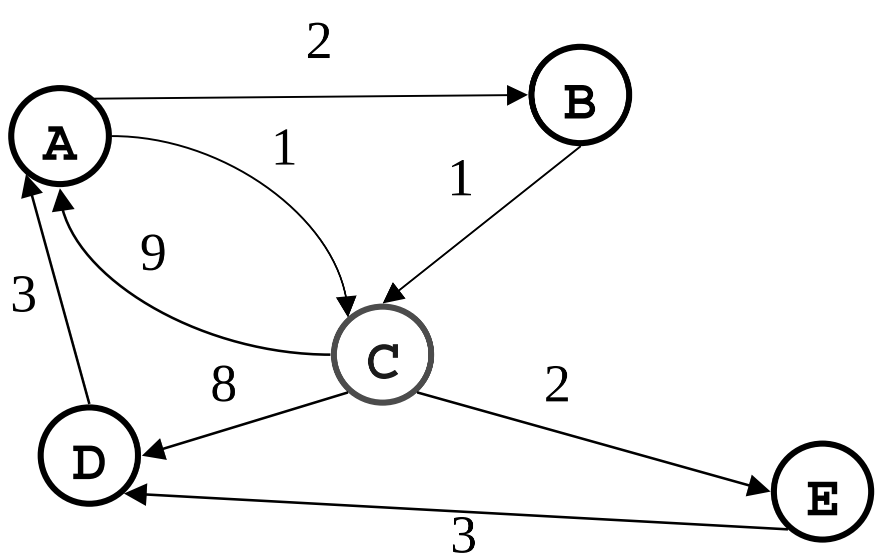
Answer:

Initial Table

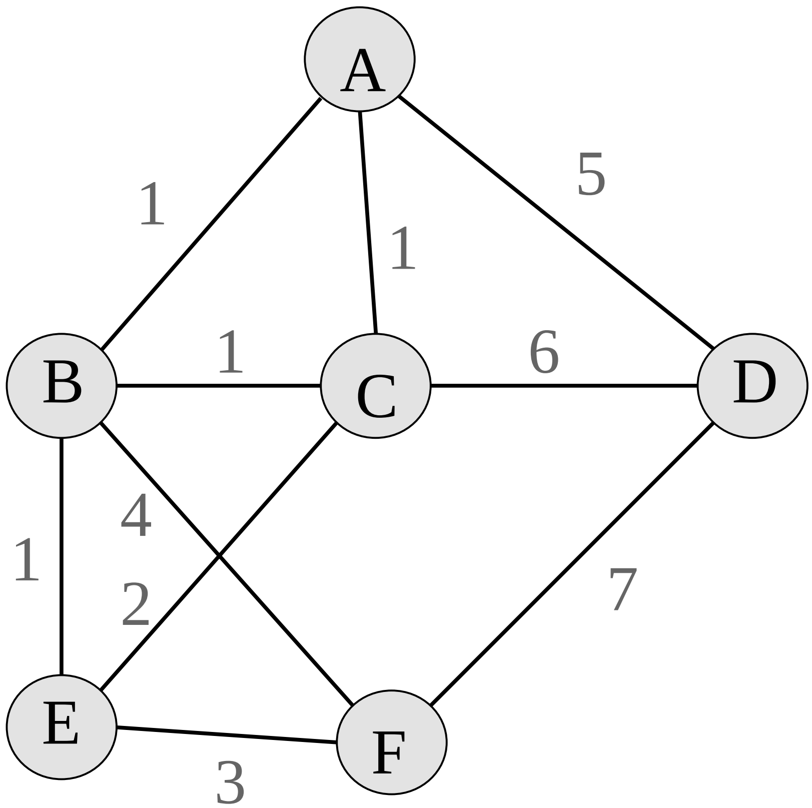
Vertex v	known	Cost d_v	Prev p_v
A	No	∞	NULL
B	No	∞	NULL
C	Yes	0	NULL
D	No	∞	NULL
E	No	∞	NULL

Final Table

Vertex v	known	Cost d_v	Prev p_v
A	Yes	8	D
B	Yes	10	A
C	Yes	0	NULL
D	Yes	5	E
E	Yes	2	C



9. **Shortest Path and MST (continued)** (a and b: 6 and 4 points)
   a. Let T be an initially empty tree that will hold the edges in a minimal spanning tree. Write down the sequence of edges (u,v) that are added to T by Kruskal's algorithm for finding the minimum spanning tree for the following weighted graph:



Answer: (A,B), (B,C), (B,E), (E,F), (A,D)
Other answers are also possible depending on which of the 1-cost edges are chosen.
E.g.. (A,C), (A,B), (B,E), (E,F), (A,D), but the total cost should be the same = 11

10. **NP-completeness** (a: 4 points)
   a. Dr. G. Nyess has just announced an algorithm for the Hamiltonian circuit problem that runs in time $O(N^9 \log N)$. Assuming Dr. Nyess' claim is true, which of the following statements is/are true (write down all that you think are true):
      i. The Hamiltonian circuit problem is no longer in NP
      ii. The Hamiltonian circuit problem is now in P
      iii. The Hamiltonian circuit problem is no longer NP-complete
      iv. P = NP
      v. $P \neq NP$

Answer: ii and iv (If any NP-complete problem can be solved in polynomial time, all problems in NP can be solved in polynomial time (they are in P) and thus, P=NP).