# King Fahd University of Petroleum & Minerals

College of Computer Science and Engineering
Information and Computer Science Department
First Semester 141 (2014/2015)

ICS 202 – Data Structures
Final Exam
Tuesday, January 6th, 2015
Time: 120 minutes

Name: ______________________________________________

| ID# | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| Section 01<br><br>Dr. Sami |
| Section 02<br><br>Dr. Ramadan |

| Question # | CLO | Max Marks | Marks Obtained |
|---|---|---|---|
| 1 | | 20 | |
| 2 | | 30 | |
| 3 | | 25 | |
| 4 | | 25 | |
| | | | |
| Total | | 100 | |

## Instructions

1. Write your name and ID in the respective boxes above and circle your section.
2. This exam consists of 9 pages, including this page, plus one reference sheet, containing 4 questions.
3. You have to answer all 4 questions.
4. The exam is closed book and closed notes. No calculators or any helping aids are allowed.
5. Make sure you turn off your mobile phone and keep it in your pocket if you have one.
6. The questions are not equally weighed.
7. The maximum number of points for this exam is 100.
8. You have exactly 120 minutes to finish the exam.
9. Make sure your answers are readable.
10. If there is no space on the front of the page, feel free to use the back of the page. Make sure you indicate this in order not to miss grading it.

**Q.1 [20 points] Multiple Choice Questions: Mark the best answer for each question below. Note: only one choice should be chosen.**

1. Consider the following code segment

```
for (int i=1, sum=0; i<n; i*=2)
    sum+=4;    // Statement 1
```

   The complexity of the above code segment is

   a. O ($n^2$)
   b. O (n log n)
   c. O (n)
   [✓ d. O (log n)]
   e. none of the above.

   log(n-1)+1

2. Consider the following code segment

```
 for (int i=1, sum=0; i<n; i+=2)
       for (int j=1, sum=0; j<n; j*=2)
           sum+=4; // Statement 1
```

   The complexity of the above code segment is

   a. O (log n)
   [✓ b. O (n log n)]
   c. O (n)
   d. O ($2^n - 1$)
   e. none of the above.

   (log(n-1)+1)*(n-1)/2

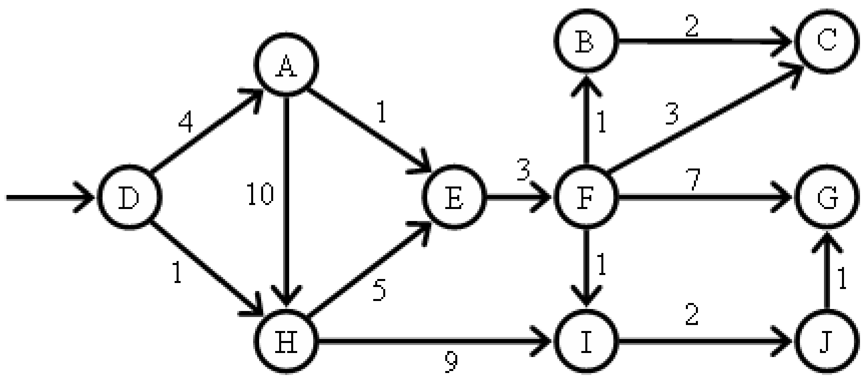
3. The big-O notation
   a. Can compare algorithms in the same complexity class
   [✓ b. Is an upper bound on the asymptotic complexity of the program]
   c. Is a bottom bound on the asymptotic complexity of the program
   d. Provide a measure which is valid for different operating systems, compilers and CPUs.

4. Among the following operations which are more efficient in the doubly linked list compared to singly linked list:
   a. Append
   b. Prepend
   [✓ c. Extract(Object ob)]
   d. ExtractFirst

   you don't need parent reference

**Q2. [30 points] (Graphs):**

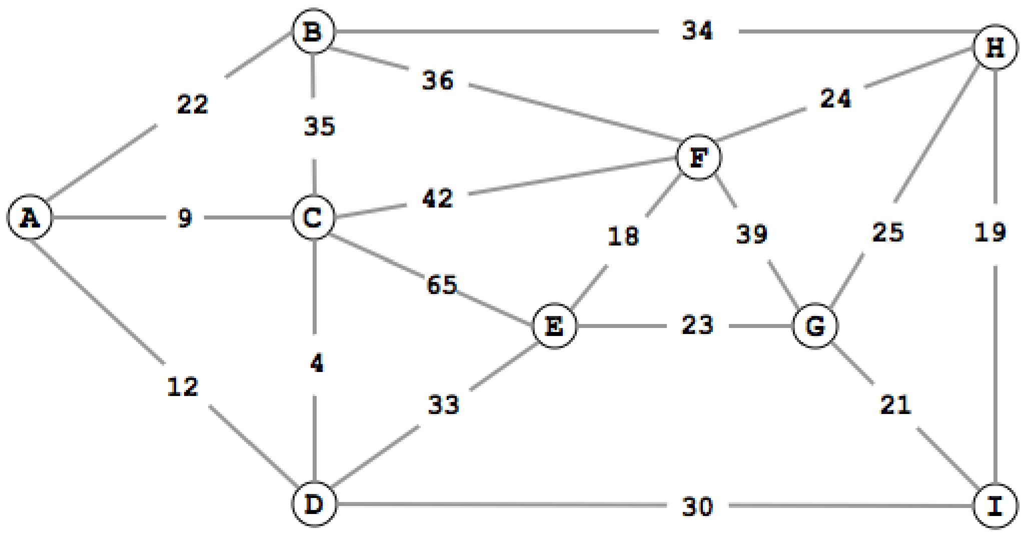
A. [10 points] Consider the following weighted directed graph G:



Apply the Dijkstra algorithm on G starting from **vertex D**:

| **Pass** | **initially** | | | | | | **weight** | **Predecessor** |
|---|---|---|---|---|---|---|---|---|
| **Active Vertex** | | D | H | A | E | F | | |
| **A** | inf | 4 | - | | | | 4 | D |
| **B** | inf | - | - | - | - | 9 | | F |
| **C** | inf | - | - | - | - | 11 | | F |
| **D** | 0 | | | | | | 0 | |
| **E** | inf | - | 6 | 5 | | | 5 | A |
| **F** | inf | - | - | - | 8 | | 8 | E |
| **G** | inf | - | - | - | - | 15 | | F |
| **H** | inf | 1 | | | | | 1 | D |
| **I** | inf | - | 10 | - | - | 9 | | H |
| **J** | inf | - | - | - | - | - | | |

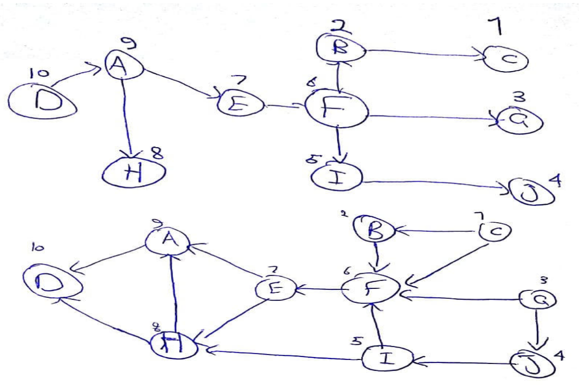
B. [10 points] Consider the following weighted undirected graph. Apply prim’s algorithm to find a minimum spanning tree of the graph starting from A. Show the minimum spanning tree.



| Pass | initially | | | | | | weight | V1 |
|---|---|---|---|---|---|---|---|---|
| Active Vertex | | | | | | | | |
| A | | | | | | | | |
| B | | | | | | | | |
| C | | | | | | | | |
| D | | | | | | | | |
| E | | | | | | | | |
| F | | | | | | | | |
| G | | | | | | | | |
| H | | | | | | | | |
| I | | | | | | | | |

C. [5 points] How many strongly connected components graph G has? List them.

$$\begin{tikzcd}
 & & A \arrow[dr, "1"] \arrow[dd, "10"] & & B \arrow[r, "2"] & C \\
{} \arrow[r] & D \arrow[ur, "4"] \arrow[dr, "1"'] & & E \arrow[r, "3"] & F \arrow[u, "1"] \arrow[ur, "3"] \arrow[r, "7"] \arrow[d, "1"] & G \\
 & & H \arrow[ur, "5"] \arrow[rr, "9"'] & & I \arrow[r, "2"] & J \arrow[u, "1"']
\end{tikzcd}$$



D. [5 points] Give the order
First.

Starting from D

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

{D}, {A}, {H}, {E}, {F}, {I}, {J}, {G}, {B}, {C}

10 SCC

**Q3. [25 points] (Hashing):**

A. [10 points] Write code to remove duplicates from an unsorted linked list using the Java predefined Hashtable structure.

```
public static void deleteDups(LinkedListNode n) {
   Hashtable table = new Hashtable();
   LinkedListNode previous = null;
```

B. [15 points] Consider inserting the following keys:

32, 33, 30, 19, 20, 25, 16, 42, 25, 29

respectively, into a hash table of size 11, using open addressing and hash function:

$$h(key)= key \% 11$$

Use double hashing as a collision resolution policy with

$$h_p(key)=key\%10$$

Show the hash table after the insertions, showing all your work.

**Q.5 [25 points]:** (**Compression**)

a) [10 points] Using Huffman coding, show the resulting Huffman coding tree for compressing the following message. Make sure you show all your work.

thisisthisisisss

b) [5 points] Compute the compression ratio, showing your work. Make sure you state any assumptions.

c) [6 points] Compress the following message using LZ78. Make sure you show all your work:

thisisthisisisisss

d) [4 points] Compute the compression ratio, showing your work. Make sure you state any assumptions.

## Quick Reference Sheet

```
public class SLLNode<T> {
    public T info;
    public SLLNode<T> next;
  public SLLNode();
  public SLLNode(T el)
  public SLLNode(T el, SLLNode<T> ptr);
}

public class SLL<T> {
    protected SLLNode<T> head, tail;
  public SLL();
  public boolean isEmpty();
  public void addToHead(T el);
  public void addToTail(T el);
  public T deleteFromHead();
  public T deleteFromTail();
  public void delete(T el);
  public void printAll();
  public boolean isInList(T el);
}

public class DLLNode<T> {
    public T info;
    public DLLNode<T> next, prev;
  public DLLNode();
  public DLLNode(T el);
  public DLLNode(T el, DLLNode<T> n,
                       DLLNode<T> p);
}

public class DLL<T> {
    private DLLNode<T> head, tail;
  public DLL();
  public boolean isEmpty();
  public void setToNull();
  public void addToHead(T el);
  public void addToTail(T el);
  public T deleteFromHead();
  public T deleteFromTail();
  public void delete(T el);
  public void printAll();
  public boolean isInList(T el);
}
```

```
public class Queue<T> {
     private …; // array or linked list
   public Queue();
   public void clear();
   public boolean isEmpty();
   public T firstEl();
   public T dequeue();
   public void enqueue(T el);
   public String toString();
}

public class BSTNode<T extends Comparable<?
super T>> {
    protected T el;
    protected BSTNode<T> left, right;
    public BSTNode();
    public BSTNode(T el);
    public BSTNode(T el, BSTNode<T> lt,
                         BSTNode<T> rt);
}

Public class Hashtable<K,V> {
void  clear()
boolean     contains(Object value)
boolean     containsKey(Object key)
boolean     containsValue(Object value)
boolean     equals(Object o)
V     get(Object key)
boolean     isEmpty()
V     put(K key, V value)
V     remove(Object key)
int   size()
String      toString()
}
```

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2,$$

$$\sum_{i=0}^{n} x^i = \frac{x^{n+1}-1}{x-1}, \quad 2^{\lg n} = n, \quad \lg ab = \lg a + \lg b, \quad \lg a^b = b \lg a$$