Boğaziçi University, Dept. of Computer Engineering

CMPE 250, DATA STRUCTURES AND ALGORITHMS

Fall 2011, Final

Name: ______________________________

Student ID: ___________________________

Signature: ___________________________

- Please print your name and student ID number and write your signature to indicate that you accept the University honour code.
- During this examination, you may not use any notes or books.
- Read each question carefully and **WRITE CLEARLY**. Unreadable answers will not get any credit.
- For each question you do not know the answer and leave blank, you can get %10 of the points, if you write only "I don't know the answer but I promise to think about this question and learn its solution".
- There are **6** questions. Point values are given in parentheses.
- You have **180 minutes** to do all the problems.

Q	1	2	3	4	5	6	Total
Score							
Max	10	10	10	20	20	30	100

1. What is .. (Give short answers. Long answers do not get any credit.)

   1.1. the notation $f(n) = \Theta(g(n))$? (2pts)

   1.2. a data structure ?

   1.3. an algorithm ?

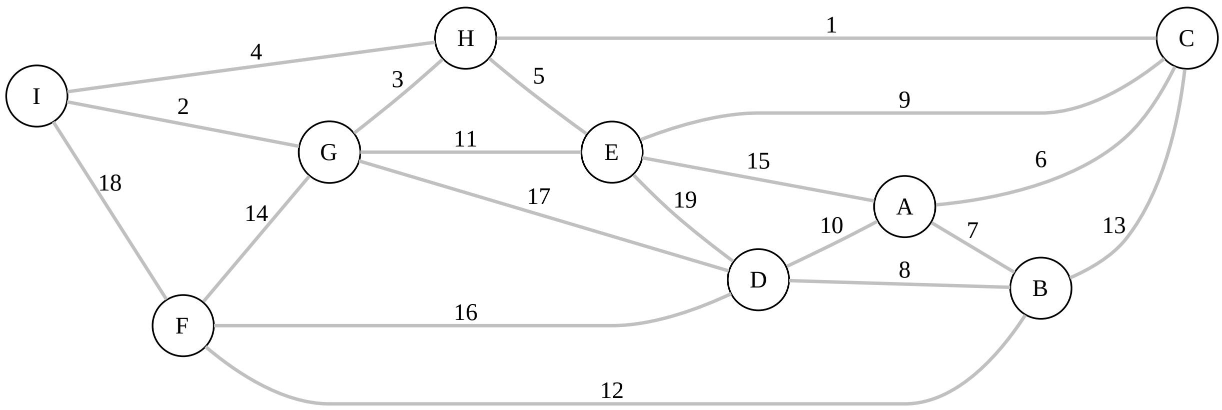
   1.4. a binary heap?

   1.5. Dijkstra's algorithm?

   (10 points)

2. (Perfect matching) Consider a bipartite graph $G = (V_1 \cup V_2, E)$ (so each edge has one endpoint in V_1 and one endpoint in V_2) and V_1 and V_2 have the same size. Describe an algorithm to find a set of edges $M \subset E$ such that M is a perfect matching, that is edges in M don't touch. *(10 points)*

3. Minimum spanning tree. For parts (a), and (b) consider the following weighted graph with 9 vertices and 19 edges. Note that the edge weights are distinct integers between 1 and 19.



3.1. Complete the sequence of edges in the MST in the order that Kruskal's algorithm includes them.

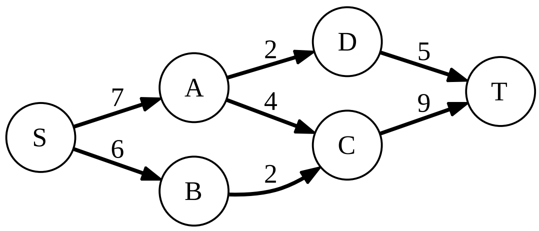
1 ____ ____ ____ ____ ____ ____ ____

3.2. Complete the sequence of edges in the MST in the order that Prims's algorithm includes them.

6 ____ ____ ____ ____ ____ ____ ____

(10 points)

4. Consider the following network (the numbers are edge capacities).



4.1. Find the maximum flow f and a minimum cut.

4.2. Draw the residual graph along with its edge capacities. In this residual network, mark the vertices reachable from S and the vertces from which T is reachable

4.3. An edge of a network is called a bottleneck edge if increasing its capacity results in an increase in the maximum flow. List all bottleneck edges in the above network.

(20 points)

5. (Sorting) Fill in the following table. (Leave empty if you are unsure as a wrong answer cancels one right answer)

	Insertion Sort	Heapsort	Mergesort	Quicksort
Sequence num (3pts each)				
Worst case time complexity				
Average case time complexity				

Below, the column on the left is the original input of strings to be sorted; the column on the right are the string in sorted order; the other columns are the contents at some intermediate step during one of the 4 sorting algorithms listed above. Match up each column by writing its number to the corresponding row labeled as 'sequence'. Use each number exactly once.

```
EII BID COP BID OYL BID
OFH COP EII COP OFH BQG
IIA EII DPD EII OFH BSH
EII EII EII EII IIA COP
NFL GMT EII GMT ODS DPD
NIF HFS BQG IIA NSR EII
LKE IIA BSH LEY NSR EII
BID IIA BID LKE EII EII
GMT ISJ EII MCF IIA EII
COP LEY EII NFL HFS EII
MCF LKE MCF NIF NQH EII
LEY MCF LEY OFH LEY GMT
NSR NFL NSR NSR NSR HFS
HFS NIF HFS HFS NFL HFS
ISJ NSR ISJ ISJ LKE IIA
IIA OFH IIA IIA EII IIA
EII BQG GMT EII EII IIA
IIA BSH IIA IIA GMT ISJ
BSH EII LKE BSH BSH LEY
HFS HFS HFS HFS COP LKE
BQG IIA NIF BQG BQG MCF
NQH NQH NQH NQH NFL NFL
ODS NSR ODS ODS MCF NFL
NSR ODS NSR NSR IIA NIF
EII EII NFL EII EII NQH
OFH OFH OFH OFH NIF NSR
EII EII EII EII EII NSR
NFL NFL NFL NFL HFS NSR
DPD DPD IIA DPD DPD ODS
NSR NSR NSR NSR ISJ OFH
EII EII OFH EII EII OFH
OYL OYL OYL OYL BID OYL
-----------------------
 U   1   2   3   4   S
```

(20 points)

6. (Dynamic Programming) Let us define a binary operation $\otimes$ on three symbols a, b, c according to the following table; thus $a \otimes b = b$, $b \otimes a = c$, and so on. Notice that the operation defined by the table is neither associative nor commutative.

$\otimes$	a	b	c
a	b	b	a
b	c	b	a
c	a	c	c

Describe an **efficient** algorithm that examines a string of these symbols, say *bbbbac*, and decides whether or not it is possible to parenthesize the string in such a way that the value of the resulting expression is a. For example, on input *bbbbac* your algorithm should return yes because $((b \otimes (b \otimes b)) \otimes (b \otimes a)) \otimes c = a$.

(30 points)