CS570
Analysis of Algorithms
Summer 2007
Exam 2

Name: _____

Student ID: _____

	Maximum	Received
Problem 1	10	
Problem 2	20	
Problem 3	20	
Problem 4	10	
Problem 5	20	
Problem 6	20	

Note: The exam is closed book closed notes.

1) 10 pts
Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

True **[TRUE/FALSE]**
Binary search could be called a divide and conquer technique

False **[TRUE/FALSE]**
If you have non integer edge capacities, then you cannot have an integer max flow

Ture [TRUE/FALSE]
The Ford Fulkerson algorithm with real valued capacities can run forever

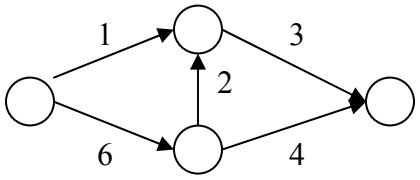
Ture [TRUE/FALSE]
If we have a 0-1 valued s-t flow in a graph of value f, then we have f edge disjoint s-t paths in the graph

Ture [TRUE/FALSE]
Merge sort works because at each level of the algorithm, the merge step assumes that the two lists are sorted

2) 20pts
Prove or disprove the following for a given graph G(V,E) with integer edge capacities Ci

a. If the capacities on each edge are unique then there is a unique min cut

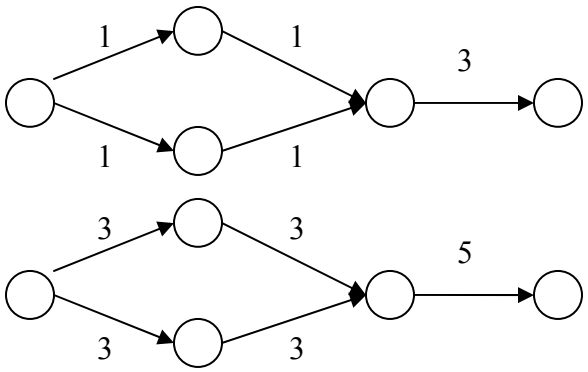
Disprove :



b. If we multiply each edge with the same multiple "f", the max flow also gets multiplied by the same factor

Prove: For each cut (A, B) of the graph G, the capacity c(A, B) will be multiplied by "f" if each edge's capacity is multiplied by "f", thus the minimal cut will be multiplied by "f" , thus the max flow also gets multiplied by "f".

c. If we add the same amount, say "a" to every edge in the graph, then the min cut stays the same (the edges in the min cut stay the same i.e.)



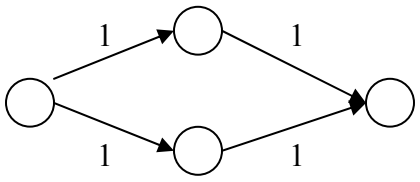
Before add 2 to every edge

After add 2 to every edge

d. If the edge costs are all even, then the max flow(min cut) is also even

The capacity of any cut (A, B) is sum of the capacities of all edges out of A, thus the capacity of any cut, including the minimal one, is also even.

e. If the edge costs are all odd, then the max flow (min cut) is also odd



Max flow: 2

3) 20 pts
Suppose you are given k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements.

(a) Here's one strategy: merge the first two arrays, then merge in the third, then merge in the fourth, and so on. What is the time complexity of this strategy, in terms of k and n?

The merge the first two array takes O(n), … the merge of the last array takes O(kn), totally there are k merge sorts. Thus, it takes $O(k^2n)$ times.

(b) Present a more efficient solution to this problem.
Let the final array to be A, initially A is empty.
Build up a binary heap with the first element from the k given arrays, thus this binary tree consists of k element.
Extract the minimal element from the binary tree and add it to A, delete it from its original array, and insert the next element from that array into the binary heap.

Each heap operation is O(logk), and take O(kn) operations, thus, the running time is O(kn*logk)

4) 10 pts
Derive a recurrence equation that describes the following code. Then, solve the recurrence equation.

```
COMPOSE (n)
    1. for i ← 1 to n
    2.         do for j ← 1 to sqrt(n)
    3.                 do print(i, j, n)
    4. if n > 0
    5.         then for i ← 1 to 5
    6.                 COMPOSE (⌊n/2⌋)
```

$T(n) = 5\ T(n/2) + O(n^{3/2})$

By the Master theorem, $T(n) = n^{\lg 5}$

5) 20 pts

Let G=(V,E) be a directed graph, with source $s \in V$, sink $t \in V$, and non-negative edge capacities $\{C_e\}$. Give a polynomial time algorithm to decide whether G has a unique minimum s-t cut. (i.e. an s-t cut of capacity strictly less than that of all other s-t cuts.)

Find a max flow f for this graph G, and then construct the residual graph G' based on this max flow f. Start from the source s, perform breadth-first or depth-first search to find the set A that s can reach, define B = V – A, the cut (A, B) is a minimum s-t cut. Then, for each node v in B that is connected to A in the original graph G, try the following codes: add v into set A, and then perform breadth-first or depth-first search find a new set A' that s can reach, if the sink t is included in A', then try next node v', otherwise, report a new minimum s-t cut. If all possible nodes v in B have been tried but no more minimum s-t cut can be found, then report the (A, B) is the unique minimum s-t cut.

20 pts
Consider you have three courses to study, C1, C2 and C3. For the three courses you need to study a minimum of T1, T2, and T3 hours respectively. You have three days to study for these courses, D1, D2 and D3. On each day you have a maximum of H1, H2, and H3 hours to study respectively. You have only 12 hours total to give to all of these courses, which you could distribute within the three days. On each one of the days, you could potentially study all three courses. Give an algorithm to find the distribution of hours to the three courses on the three days

If T1+T2+T3 > 12 or T1+T2+T3 > H1 + H2 + H3, then report no feasible distribution can be found.
Else, start C1 with T1 hours, and then C2 with T2 hours, and finally C3 with T3 hours.