BLG 372E ANALYSIS OF ALGORITHMS
FINAL – MAY 21, 2013, 9:00-11:00 AM (2hours)- PART A

A

Q1 (15pt)	Q2 (15 pt)	Q3 (20 pt)	Total (50 pt)

On my honor, I declare that I neither give nor receive any unauthorized help on this exam.

Student Signature:________________________

Write your name on each sheet. Write your answers neatly (in English) in the space provided for them. You must show all your work for credit. Books and notes are closed. Good Luck!

Q1)[15pts] Dynamic Programming

You are given a knapsack which can take a maximum of W=5 kgs. Given the items and their values and weights below, choose items to put in the knapsack so that the total value of the items you picked is maximized.

ItemID (i)	Value (v_i)	Weight (w_i)
1	1	4
2	3	2
3	4	1

Q1a)[7pts] Write down the **dynamic programming algorithm** you will use. OBJECTIVE FUNCTION

$$OPT(i,w) = \begin{cases} 0 & \text{if } i=0 \\ OPT(i-1,w) & \text{if } w_i > w \\ \max\{OPT(i-1,w),\ v_i + OPT(i-1, w-w_i)\} & \text{else} \end{cases}$$

INPUT: $N, w_1, .. w_N, v_1, .. v_N$
$M = \text{Array}[0..N, 0..W]$
for $w = 0..W$ $M[0,w] = 0$
for $i = 1..N$
  for $w = 1..W$
    if $w_i > w$ $M[i,w] = M[i-1,w]$
    else $M[i,w] = \max\{M[i-1,w], v_i + M[i-1, w-w_i]\}$
return $M[N,W]$

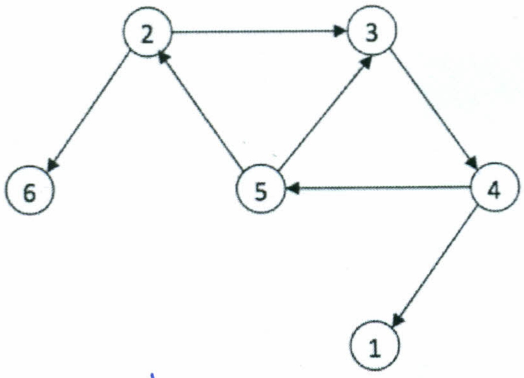
Q1b)[8pts] Compute the solution for the problem instance above. Which items did you choose?

M[i,w]	0	1	2	3	4	5
∅	0	0	0	0	0	0
{1} $w_i=4$	0	0	0	0	1	1
{1,2} $w_i=2$	0	0	3 (pick 2)	3	3	3
{1,2,3} $w_i=1$	0	4 (pick 3)	4	7 (pick 3 & 2)	7	7 (pick 3 & 2)

Items 3 and 2 have been picked

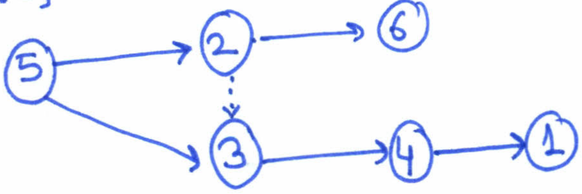
 Name and Student ID: Gataltepe

Q2) [15pts] (Graphs)



Q2a)[8pts] Given this **directed** graph, **starting from Node 5**, write down the Breadth First Search (BFS) Tree. Note that since the graph is directed, in the tree you can visit node v after node u, only if there is an edge (u,v).
Show all the steps of your work.

$L_0 = \{5\}$ $L_1 = \{2,3\}$ $L_2 = \{6,4\}$ $L_3 = \{1\}$

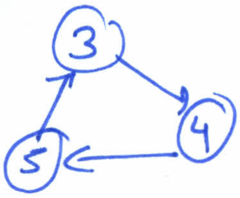


Discovered	[5]	[1]	[2]	[3]	[4]	[6]
L_0	T	F	F	F	F	F
L_1	T	F	T	T	F	F
L_2	T	F	T	T	T	T
L_3	T	T	T	T	T	T

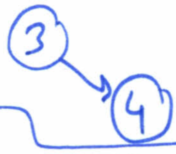
$L_4 = \emptyset$

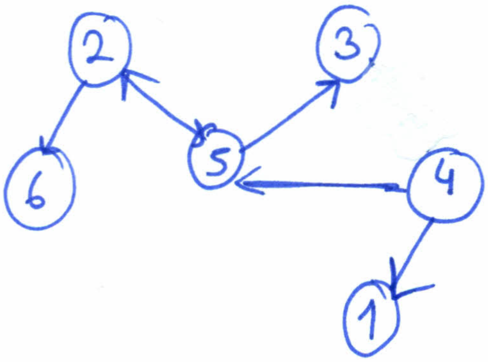
Q2b)[7pts] Why isn't there a topological ordering in this graph?
By deleting at most one edge, produce a graph that has a topological ordering and write down a topological ordering for the new graph.

Because there's a directed cycle

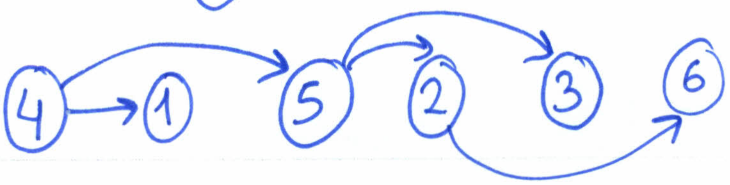


I delete edge





Topological ordering algorithm:
repeat
Find a node u with no incoming edges.
add (u,v) to the ordering.
Delete u
until there are no nodes left.



 Name and Student ID: Gatalteam

Q3)[20pts] Intractability
A supermarket is trying to find a diverse set of at least k customers so that no two customers selected in the subset have ever bought the same item. This problem is called the Diverse Subset (DIV-SUBSET) Problem and an instance of it is given below. In this problem instance, if a customer i has bought item j, the table entry at (i,j) is 1, otherwise it is 0.

S

S	Detergent (d)	Milk (m)	Cherry (c)	Bread (b)
Ahmet (A)	0	1	0	1
Bahri (B)	1	1	0	0
Cemal (C)	0	0	0	1

Q3a) [8pts] Show that the DIV-SUBSET problem is NP-Complete, by computing a polynomial-time reduction which can convert **any given instance** of the DIV-SUBSET Problem to an instance of the INDEPENDENT-SET problem, or any one of the other NP Complete problems (3-SAT, VERTEX-COVER, SET-COVER, CIRCUIT-SAT) we have seen in the class.

Let $S(u,j)$ show if customer u bought item j.
Construct a graph $G=(V,E)$ as follows.
n .... $V = \{$ set of customers $\}$
$E = \{(u,v)$: customer u and v have bought the same item,
i.e $\exists j,\ S(u,j)=1$ and $S(v,j)=1$
↑ products.

If there is a DIV-SUBSET of size k,
then there is an independent set of size k

Q3b) [3pts] Write down the worst-case complexity of your polynomial-time reduction algorithm in terms of m (no of customers), n (no of products) and k (the subset size), explaining why:
O(..mn^2.......)

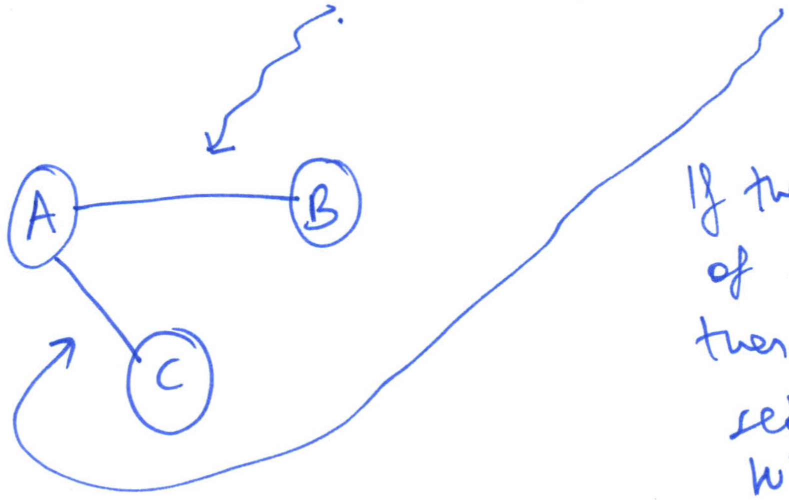
To construct E,

m { for each item j
  V_j: set of nodes to be connected because of the item
  $V_j = \phi$
  n { for each customer u
    if u bought j, $V_j = \{V_j \cup \{u\}$
  n^2 { connect all the nodes in V_j.

worst case
$m*(n+n^2) \approx O(mn^2)$

Q3c) [4pts] Apply the polynomial-time reduction that you have found to the example DIV-SUBSET problem instance below.

	Detergent (d)	Milk (m)	Cherry (c)	Bread (b)
Ahmet (A)	0	1	0	1
Bahri (B)	1	1	0	0
Cemal (C)	0	0	0	1



If there's an indep-set of size k, then there's a set of k customers who haven't ever bought the same item.

Q3d)[5pts] Show that 3-SAT problem is in NP.

NP: Non deterministic Polynomial. Given a solution for 3-SAT we need to show that there's a certifier which can check the solution in polynomial time in the number of clauses in the 3-SAT formula

$\Phi = C_1 \wedge C_2 \wedge .. \wedge C_n$, C_i = A clause Disjunction of 3 literals.

S: Solution: An assignment of variables $x_1, .., x_m$

Cert (Φ, S)

for $i = 1..n$
 plug in each literal x in clause c_i, evaluate C_i
 if C_i = False
 return false
return TRUE

$O(n \times 3)$
$\approx O(n)$

Since the certifier runs in poly time, 3-SAT is in NP