100 marks **CS 601 Midsemester Test** **2:00-4:00, 18/9/10**

You may reduce from any NP-hard problems that we have studied, e.g. Circuit-SAT, Independent Set, Vertex Cover, Set Cover/Simplified facility location, Subset Sum, Partition, Graph Colouring, Clique.

Problem 1: While we have noted in class that it is NP-complete to find the minimum vertex cover for a general graph, it turns out that for the special case of bipartite graphs, the vertex cover can in fact be found in polynomial time, as you will see in this problem.

(a)[5 marks] Show that the size of the minimum vertex vertex cover for a bipartite graph $G = (U, V, E)$ is at least the size of the maximum matching in G. *Hint: How many vertices are needed to cover just the edges in the matching? Explain properly.*

Let V' denote a vertex cover. Every edge in the graph must be incident on at least one of the vertices in V'. Since a matching is a subset of all edges, every edge in a matching in the graph must also be incident on one of the vertices in V'. Note now that by the definition of a matching, two edges in a matching are incident on distinct vertices. So the edges in a matching M must edges must be incident on $|M|$ different vertices in V'. Thus there must be at least M vertices in V'. Thus we have $|V'| \geq |M|$, for any vertex cover V' and any matching M. But this applies to the minimum vertex cover and the maximum matching as well.

(b)[5 marks] Suppose we form a flow network from the bipartite graph G in the usual manner, i.e. add vertices s, t with edges from s to all $u \in U$ and to t from all $v \in V$, with all edge capacities 1, and all edges oriented from s to U to V to t. Show that there exists a min-cut (A, B) where $s \in A, t \in B$ and there are no edges leaving A and entering $V - A$. By $V - A$ we mean the set obtained by removing $V \cap A$ from V.

Suppose there are x edges leaving A and entering $V - A$. These edges can be incident on some y vertices from $V - A$. These vertices will have exactly y edges leaving to s. So if we include these y vertices into A, the new cut size will decrease by x and increase by y. But $y \leq x$, and hence the new cut will be a min cut given that the old one was.

Note that there can be cuts that do not have the property. So the entire argument is needed.

(c)[10 marks] Using the min-cut as found above, find a vertex cover for G of size same as the min-cut.

VC = $(U - A) \cup (A \cap V)$. 4 marks

The edges to be covered are:

$U \cap A$ to $A \cap V$: covered by $A \cap V$ 2 marks.

$U \cap A$ to $V - A$: do not exist by definition. 2 marks

$U - A$ to V: covered by $U - A$. 2 marks.

(d)[5 marks] Show (A,B) and the vertex cover for the following graph: U, V are numbered 1-5, 6-10 top to bottom, edges are (1,6), (2,6), (3,6),(3,7), (4,7),(4,8),(4,10), (5,8),(5,9),(5,10). Draw the graph and mark the required sets in it. You get no credit for (b),(c) unless you do this part.

A = s,1,2,3,6,7. 2 marks

VC = 4,5, 6,7. 3 marks

Problem 2: In this problem you are to prove the relationship between the decision and construction version of the graph colouring problem. Just to remind you, the problems are as follows. Let $G = (V, E)$ be an undirected graph. A (vertex) k colouring of G is an assignment of an integer c_u to every vertex u such that $1 \leq c_u \leq k$, and forall $(u, v) \in E$, we have $c_u \neq c_v$.

Let CGC denote the problem, given a graph G, of colouring it using the minimum number of colours k. Let DGC denote the problem of deciding, given a graph G and integer k, whether G has a k colouring.

(a)[5 marks] Show that $DGC \leq_{Karp} CGC$. It is enough if you just give the instance map and solution map functions F, G.

F(G,k){ Return G. }

G(answer,G,k){ if answer $\leq k$ return true, else false.}

5 marks

(b)[5 marks] Suppose a graph G can be coloured in k colours. Suppose no edge can be added to G while maintaining the property that it can be coloured in k colours. Give a simple algorithm for colouring G, with a brief explanation.

If two vertices are not connected, they must have the same colour, else an edge could have been added. 3 marks So pick any vertex, and assign it and vertices it is not connected to the first colour. And so on. 2 marks

(c)[10 marks] Show that $CGC \leq_{Cook} DGC$. *Hint: Use the result of part (a). You may assume the existence of the algorithm even if you didnt solve (a).*

By DGC(G,i) we mean the solution to the DGC instance (G, i).

F(G){

1. In the range $k = 1$ to $|V|$ find minimum k such that DGC(G,k)=true.

2. Return $F'(G, k)$.

}

$F'(G, k)${

1. For each edge i not in G: // earlier I had written "in G"
   (a) let $G = G$ with e_i added.
   (b) If $DGC(G, k) == false$, then remove e_i from G.
2. At this point we have a graph in which no edge can be added without increasing the number of colours. So colour it in k colours.

}

2 marks to find the minimum number of colours. 6 marks to get the basic idea of adding edges. 2 marks to argue that there are only polynomial calls to DGC.

Problem 3:[20 marks] Consider the load balancing problem done in class. The input to this is a bipartite graph $G = (U, V, E)$ where U is the set of jobs, V is the set of processors. There is an edge from $u \in U$ to $v \in V$ if job u can be scheduled on processor v. For each job u we are also given the time t_u (integer) needed for its execution. We are required to assign jobs to processors such that maximum load of any processor is as small as possible, where the load of a processor is the total time of jobs assigned to it. For the decision version, we are also given at target time T, and the goal is to decide if there exists a schedule with maximum load T. Show that this problem is NP-hard.

Reduce from Partition. Instance of partition: $S = x_1, \ldots, x_n$.

F(S){

1. For each x_i construct vertex i in U. Let $t_i = x_i$. 4 marks
2. $V = \{a, b\}$ consists only of 2 vertices. 4 marks.
3. There is an edge from every $u \in U$ to every $v \in V$. 4 marks.
4. $T = (\sum_i x_i)/2$. 4 marks.
5. Return the instance $(G = (U, V, E), \{t_u\}, T)$ that we have constructed.

}

claim 1: Instance can be constructed in polytime. 1 marks.

claim 2. If S has a partition, then tasks can be scheduled in T.

proof: Let S', S'' denote the partition. If $x_i \in S'$, then schedule $i \in U$ on a, else on b. Since there is a complete graph from U to V, this is allowed. Since $\sum_{i \in S'} x_i = T/2$, the total load given to a is also T. Likewise for b. 2 marks.

claim 3: If tasks can be schedule in T, then S has a partition.

proof: If $i \in U$ is assigned to a, then include it in S' else in S''. 1 marks.

Problem 4:[15 marks] Show that the decision version minimum independent set problem reduces to Circuit Satisfiability. To remind you the independent set problem, IS is: Given an undirected graph G and integer k, does G have an independent set of size k, i.e. a subset V' of k vertices no two of which are connected by the same edge.

F(G,k){

The circuit will have $n = |V|$ inputs, input x_i corresponding to vertex i. 2 marks.

For each edge $e = (i, j)$ put AND gate G_e with x_i, x_j as inputs. To the output of G_e connect a NOT gate N_e. 3+2 marks.

Connect a circuit which counts the number of 1s in all x_is and connect the output of that to a comparator circuit the other input being hardwired to the number k. Let X denote the output of the comparator which is 1 if the inputs to the comparator are equal. 2 marks

Connect the outputs of all N_e gates and X to an AND gate. 1 mark.

The output of this and gate is the circuit output.

}

Claim 1: The circuit can be constructed in polytime. 1 mark

Claim 2: If G has a size k independent set V', then the circuit is satisfiable.

Proof: If vetex $i \in V'$, then set $x_i = 1$ otherwise 0. Since $|V'| = k$, clearly X in the circuit will be 1. Since V' is independent, for any edge (i, j) both x_i, x_j will not be 1. Hence $G_{(i,j)} = 0$. Hence $N_{(i,j)} = 1$. Thus the output of the circuit will be 1. 2 marks

Claim 3: If circuit is satisfiable, then there exists an independent set V' of size k.

Proof: Consider the satisfying assignment. In this if $x_i = 1$, then we include i in V', otherwise not. Since the circuit is satisfiable, X must be 1, i.e. the comparator has output true, thus exactly k x_is must be 1, hence exactly k vertices are in V'. Also, each N_e is 1, hence each G_e must be 0. Thus at least one of x_i, x_j must be 0, where $e = (i, j)$. Thus at least one endpoint of each edge (i, j) is not in V'. Thus V' is independent. 2 marks.

Problem 5: In this you will partially prove the correctness of a reduction from vertex cover to subset sum. The instance for VC is an undirected graph $G = (V, E)$ and an integer k. The instance for the partition problem is a set $S = \{s_1, \ldots, s_p\}$ of integers, and a target integer t, all represented as w bit integers. Here is the instance map function:

F(G,k) {

1. Number the edges of G from 0 to $m - 1$, and the vertices from 0 to $n - 1$.
2. Associate a weight $w_i = 10^i$ with edge numbered i.
3. For each vertex u construct $s_u = 10^m + \sum_v w_{(u,v)}$, i.e. the total weight of the edges incident on u.
4. Also construct $s_{n+i} = w_i$, for all $i = 0, \ldots, m - 1$.
5. Let $t = k \cdot 10^m + \sum_{i=0}^{m-1} 2 \cdot 10^i$
6. Let $w = 4m + 4n$.
7. Return $S = \{s_0, \ldots, s_{n+m-1}\}, t$.

}

(a)[10 marks] Consider a VC instance with vertices 0,1,2,3, and edges (0,1), (1,2), (2,0), (2,3) numbered respectively 0,1,2,3. Also let $k = 2$. Show $F(G, k)$. It might be convenient to write the numbers in decimal.

$s_0 = 10101$, $s_1 = 10011$, $s_2 = 11110$, $s_3 = 11000$, $s_4 = 1$, $s_5 = 10$, $s_6 = 100$, $s_7 = 1000$.

$T = 22222$

(b)[5 marks] Suppose some subset S' of S (not just for the above instance, but in general) adds up to at least $k \cdot 10^m$. What can you say about S'? Why?

S' must contain at least k of the numbers between s_0 and s_{n-1}. 2 marks This is because carries from the preceding digits will not reach digit m. 3 marks

(c)[5 marks] This VC instance has a yes answer, e.g. {0,2} is a vertex cover of size 2. Can you write down the "corresponding" solution to the SS problem?

$s_0 = 10101$, $s_2 = 11110$, $s_4 = 1$, $s_5 = 10$, $s_7 = 1000$.