Major, CSL630, 120 mins, 60 marks, Entry No. ____________________
Please write answers in the space provided. No extra sheet will be given.

Some coins are spread in the cells of a $n \times m$ board, one coin per cell. A robot, located in the upper left cell of the board, needs to collect as many of the coins as possible and bring them to the bottom right cell. On each step, the robot can move either one cell to the right or one cell down from its current location. When the robot visits a cell with a coin, it picks up that coin. Devise an efficient algorithm to find the maximum number of coins the robot can collect and a path it needs to follow to do this. (10)

Let A be an $n \times m$ matrix and define

2 marks [$A[i,j]$ = maximum number of coins robot can pick up in going from $(1,1)$ to (i,j)

Then

4 marks [$$A[i,j] = \max(A[i-1,j], A[i,j-1]) + C[i,j]$$

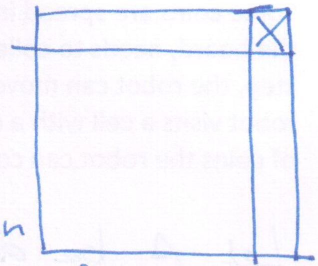
where $C[i,j] = \begin{cases} 1 & \text{if cell } (i,j) \text{ contains coin} \\ 0 & \text{otherwise} \end{cases}$

2 marks [The matrix A can be computed in $O(mn)$ time since it takes $O(1)$ time to fill each cell

2 marks. [$A[m,n]$ contains the solution to the problem.

n^2 different numbers are written on n^2 cards, one number per card. The cards are arranged in *n* rows and *n* columns, in increasing order in each row (left to right) and each column (top down). All the cards are turned faced down so that you cannot see the numbers written on them. Devise an algorithm to determine whether a given number is written on one of the cards by turning up less than *2n* cards? (10)

Solution: Consider the element at location $(1, n)$, say this is p.



Suppose we are searching for x.

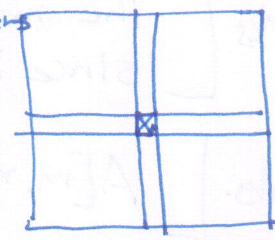
If $x < p$ then x cannot be in the last column since all elements there are larger than p. In this case eliminate last column.

If $x > p$ then x cannot be in the first row since all elements in 1^{st} row are less than p. Eliminate first row.

If $x = p$ done. STOP.

By opening one card we eliminate either one row or one column. Hence only $2n$ cards need to be checked.

Common mistakes: If you open the middle element $(\frac{n}{2}, \frac{n}{2})$ then you will still have to solve three subproblems in arrays of size $n/2 \times n/2$. So the recurrence you get will be



$$T(n) = 3T(\tfrac{n}{2}) + 1$$

whose solution is $n^{\log_2 3}$

Marking scheme: · 2 marks for an $O(n \log_2 n)$ solution

· 2 marks if you opened the $(n/2, \frac{n}{2})$ element & said that you need to solve 3 subproblems of size $n/2 \times n/2$.

The 52 playing cards in a deck are arranged randomly, face up, to form an array with 4 rows and 13 columns. Show that it is always possible to pick one card from each column so that the 13 cards you get are all different in value i.e. ignoring the suit you get all the cards ace, king, queen, jack, 2, 3, 4, 5, 6, 7, 8, 9, 10. (10)

Create a bipartite graph $G = (U, V, E)$. For every column we have a vertex in U & for every value a vertex in V.
Hence $|U| = |V| = 13$.

The edge $(u, v) \in E$ if column u contains the card with value v. Since every column has 4 cards ~~&~~ or every value, there are 4 cards of that value, the degree of every vertex is 4. Hence the graph is 4-regular.

~~A~~ The graph G has a perfect matching. For contradiction assume that it does not. Hence there is a Hall set, a set $S \subseteq U$ s.t. $|N(S)| < |S|$. Now there are $4|S|$ edges incident to vertices in S & all these edges are also incident to vertices in $N(S)$.
So $4|S| \geq 4N(S)$ & hence the contradiction.
The edges of the perfect matching give the card to pick from each column

Marking scheme: 4 marks for constructing the bipartite graph or an equivalent flow network

+4 marks for ~~claiming that this graph has a perfect matching~~ arguing that this is a 4-regular graph.

Common mistakes:

1) building a flow-network & saying (without proof) that it will have a flow of 13 units.

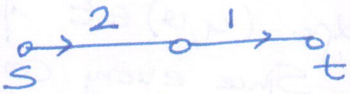
2) going column by column to pick a card. No such approach is likely to work without backtracking. Proving that by backtracking you will succeed is even more difficult.

Decide whether the following statements are true or false. If true, give a short explanation. If false, give a counterexample. Let G be a flow network with a source s, a sink t, and a positive integer capacity c_e on every edge e.

[5 marks] If f is a maximum $s-t$ flow in G then f saturates every edge out of s with flow (that is for all edges e out of s, we have, $f(e) = c_e$).

[2 marks] FALSE

3 marks – counterexample

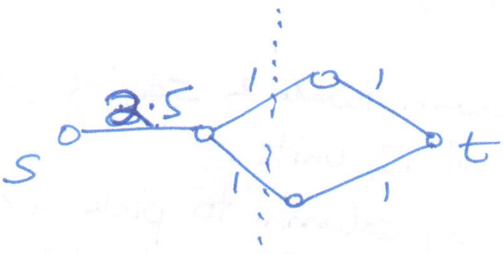


maxflow = 1 does not saturate edge out of s.

[5 marks] Let (A, B) be a minimum $s-t$ cut with respect to these capacities. Suppose we add 1 to every edge capacity. Then (A, B) is still a minimum $s-t$ cut with respect to these new capacities.

2 marks : FALSE .

3 marks : COUNTER EXAMPLE



mincut has capacity 2.

After every edge capacity increases by 1, the mincut changes & now has capacity 3.5

A *cycle cover* in a directed graph *G=(V,E)* is a subset of vertices, $V' \subset V$, such that every cycle of *G* passes through a vertex in V'. Given a directed graph *G* and an integer *k* the *cycle cover problem* is to determine if there is a cycle cover of size at most *k*. Show that this problem is NP-complete by reducing the vertex-cover problem to it.(10)

To prove that the problem is in NP.

- a certificate for ~~the~~ a 'YES' instance is a subset, say S, of at most k vertices which forms a cycle cover [2 marks]
- the certifier needs to check in polynomial time that S is indeed a cycle cover. This can be done by removing S & incident edges & checking that the remaining graph is acyclic (using DFS as done in class). [2 marks]

To prove the problem is NP-hard.

Let $(G=(V,E), k)$ be an instance of the vertex cover problem.

Construct $G' = (V, E')$ by replacing each edge $e=(u,v) \in E$ with two directed edges $e_1 = (u,v)$ & $e_2 = (v,u)$ [2 marks]

Claim: G' has a cycle cover of size k iff G has a vertex cover of size k.

Proof. G' has cyclecover of size $k \Rightarrow G$ has vertex cover of size k.

Let S be a cycle cover in G' having k vertices. In G' whenever we have an edge (u,v) we also have edge (v,u). Hence there is no edge between any two vertices in $V-S$ because then there would also be the reverse edge & therefore a cycle. So $V-S$ is an independent set in G & hence S is a vertex cover in G [2 marks]

G has a vertex cover of size $k \Rightarrow G'$ has a cycle cover of size k

Let S be a vertex cover in G. Since $V-S$ is independent in G, in G' there is no edge between any two vertices in $V-S$. Hence S is a cycle cover in G'. [2 marks]

COMMON MISTAKE

Reducing cycle cover to vertex cover.

Consider the following 3-partition problem. Given integers $a_1, a_2, \ldots a_n$, we want to determine whether it is possible to partition $\{1,2,\ldots,n\}$ into three disjoint subsets I,J,K such that

$$\sum_{i\in I} a_i = \sum_{j\in J} a_j = \sum_{k\in K} a_k = \frac{1}{3}\sum_{i=1}^{n} a_i$$

For example, for input (1,2,3,4,4,5,8) the answer is *yes* because there is the partition (1,8), (4,5), (2,3,4). On the other hand, for input (2,2,3,5) the answer is *no*. Give an efficient algorithm for this problem. (10)

We solve this by Dynamic Programming. Let X be a 3-dimensional array of size $n \times W \times W$ where $W = \frac{1}{3}\sum_{i=1}^{n} a_i$.

Define $X[i,j,k] = 1$ if $\exists\ S_1, S_2 \subseteq \{a_1, a_2 \ldots a_i\}$, $S_1 \cap S_2 = \phi$ and $\sum_{a_i \in S_1} a_i = j$ & $\sum_{a_i \in S_2} a_i = k$.

$= 0$ otherwise.

Now,

$X[i,j,k] = 1$ if $X[i-1,j,k] = 1$ OR $X[i-1, j-a_i, k] = 1$ OR $X[i-1, j, k-a_i] = 1$.

Hence the entire array can be filled in time $O(nW^2)$.

Finally, the answer to the 3-partition problem is YES if $X[n, \frac{W}{3}, \frac{W}{3}] = 1$

COMMON MISTAKES: Many students have given the following solution:

Find a subset S whose sum is W, remove these integers & find another subset of sum W.

For the instance (1,2,3,4,4,5,8) if the first subset found is (1,3,5) then the remaining integers are (2,4,4,8) & there is no subset here of sum 9. However the original instance is a YES instance as shown above (in the question).

There is no way of formulating this as a max flow problem. Some students have created networks with edge capacity equal to the given integers. There is no way to ensure that these edges would be saturated.

MARKING SCHEME:

2 marks given if you gave the "wrong solution" described above i.e. find a set S of sum W, remove it & repeat.