Honor code :-

I hereby affirm that I will not plagiarize, use unauthorized materials, or give or receive illegitimate help on the final examination for CSOR 4231.

I affirm that I have reviewed and will adhere to the Academic Honesty Policies of the Computer Science Department mentioned in the course syllabus.

Date:- 12th May, 2022

Name:- Chandan Suri

Signature:- Chandan Suri

1.

Chandan Suri
CS4090

(1) FALSE

(2) TRUE

(3) FALSE

(4) TRUE

Chandan Suri
CS4090

2.

(a) FALSE,

Recasting any LP problem such that :-

→ Either max/min problem

→ All constraints are inequalities in same dir^n

→ All vars are non-ve.

→ we can convert it to a constraint with 2 inequalities.

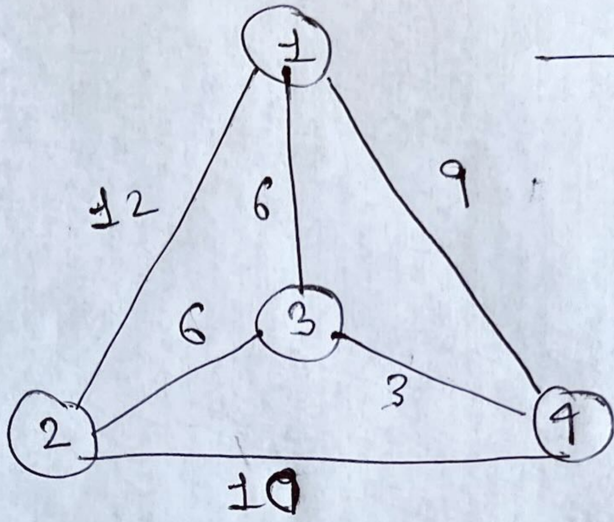
∴ Reverse is not true.

(and we cant set inequality with the signs here, we will have to introduce one more inequality / equality for this).

Hence, not true

2. b) TRUE,

Chandan Sur
CS4070



→ for this graph here, metric TSP holds if

$d_{ij} \le d_{ik} + d_{kj}$

(this is a complete graph)

To prove ⇒ ∃ 2 approx algo to prove metric TSP.

Assume ⇒

if $A(G') \le 2n$, then G has a Ham. cycle

if $A(G') \ge 2n+1$, then G does not have a Ham. cycle

As approx algo is efficient, we can't assume that we can't get that.

This is only possible when P = NP.

Chandan Suri - CS4990

(3.) Given:- $n = 13$ entries

$k = 3$

$h_1(x) = (x+3) \bmod 13$

$h_2(x) = (2x+2) \bmod 13$

$h_3(x) = (3x+1) \bmod 13.$

(a) for a given set $S = \{6, 9, 15\}$.

first, we need to construct the bloom filter.

for $x = 6$,

$$\begin{cases} h_1(6) = (6+3) \bmod 13 = 9 \\ h_2(6) = (12+2) \bmod 13 = 1 \\ h_3(6) = (19) \bmod 13 = 6 \end{cases}$$

for $x = 9$,

$$\begin{cases} h_1(9) = (9+3) \bmod 13 = 12 \\ h_2(9) = (20) \bmod 13 = 7 \\ h_3(9) = (28) \bmod 13 = 2 \end{cases}$$

for $x = 15$,

$$\begin{cases} h_1(15) = 18 \bmod 13 = 5 \\ h_2(15) = 32 \bmod 13 = 6 \\ h_3(15) = 46 \bmod 13 = 7 \end{cases}$$

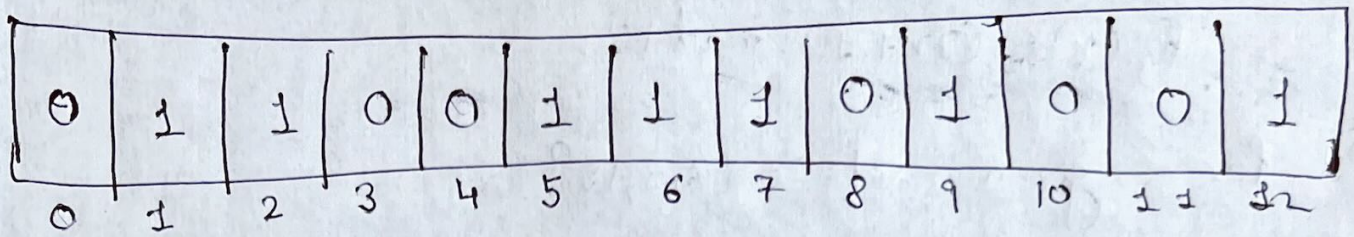
∴ The bloom filter looks like the one below:-

Treating the values out of the hash function as indices for the bloom filter:—

Chandan Suri
CS4090

(Initially all values in B are initialized to zeros).

B ⇒



Arr.

(b)

(i) Is $x = 8 \in S$?

Computing values from hash functions:—

$h_1(8) = 11 \bmod 13 = 11$

$h_2(8) = 18 \bmod 13 = 5$

$h_3(8) = 25 \bmod 13 = 12$

$B[11] = 0$, $B[5] = 1$, $B[12] = 1$

∴ "No" is the answer field ∴ $B[11] = 0$.

(even if one of the values becomes zero, then we say that 8 does not belong to the set S).

(ii) Is $x = 16 \in S$?

$h_1(16) = 6$; $h_2(16) = 8$; $h_3(16) = 10$.

"No" is the answer here as $B[8] = B[10] = 0$.
As these are zeros, ∴ this query means that 16 does not belong to the set S.

Chandan Suri
CS4090

(iii) "Is $x = 19 \in S$"?

$h_1(19) = 9$; $h_2(19) = 1$; $h_3(19) = 6$.

"Yes" is the answer here, as $B[9] = B[1] = B[6] = 1$.

This means that this query returns 1 and also that 19 does not belong to the set S.

Ans

(4) (a)

Decision version:-

Given $(n, \{v_1, \ldots, v_n\}, \{w_1, \ldots, w_n\}, W)$ and a value k, is there a subset of the items with total weight $\leq W$ and sum of all values of the items chosen $\geq k$? (at least k : this is a maximization problem).

(5)

→ Knapsack is NP?:-

→ If we have a certificate to which is subset, max weight & value k.

4. (b)

integer weight $w > 0$,
items = n ($\forall$ item i, $\exists$ integer weight $w_i > 0$ and value $v_i \geq 0$)

Subset S should contain the items that maximizes the total value of S.

Consider the constraint of the total weight of items in $S \leq W$.

Since, we want the item either to be picked or not, we can consider ~~this an integer programming problem.~~ x_i to be integer. but find value is real valued

Thus, I am formulating an LP solution for Knapsack:- (linear programming).

Variables:- Introducing one binary variable x_i for every item i, such that

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is included in the Subset} \\ 0, & \text{otherwise} \end{cases}$$

Constraints:- Sum of weights of the selected items $\leq W$

$$\sum_{i \in \text{items that maximize the total value of S.}} w_i \leq W$$

Objective function: (LP formulation).

As we need to maximize the value of the selected items,

LP for Knapsack:-

(if n items).

$$\max \sum_{i=1}^{n} v_i x_i$$

Chandan Suri
CS 4090

Subject to :- (constraints)

$$\sum_{i=1}^{n} w_i x_i \leq W$$

where $x_i \in \{0, 1\}$

for every $1 \leq i \leq n$

This is the IP formulation for Knapsack.

Ans

5.

Chandan Suri
CS420

→ Algorithm:

There are n containers and m trucks. As each truck j can hold upto k containers.

For every container i, we find a subset S_i of trucks that may carry the container.

Determining → load all n containers in m trucks s.t. no truck is overloaded. (whether we can do this or not!)

→ we will solve this by forming a bipartite graph. On LHS we have n containers that need to be matched to m trucks. (on RHS).
We form directed edges going from the containers to the trucks (in RHS).
Each container i will have all possible directed edges going to all the trucks in the subset S_i of truck allowed to carry that container.

→ Next, we will build a flow network for this by adding a source (s) and a sink node (t). We connect source to all the containers with edges from source to each container and set its capacity to 1.
(all)
We all set the capacity of each edge from container to the trucks to 1.

Then, we will form edges (directed) going from each truck to the sink node and we will set its capacity to k. (∵ each truck can carry at most k containers.

Chandan Sura
cs4020

⇒ Now, we will run ford-fulkerson Algorithm on this flow N/w. If we find the max flow to be n, we can conclusively say that its possible to load all n containers such that no truck is overloaded.

Complexity of

ford-fulkerson Algo complexity is $O(mnU)$ where U is the largest capacity of an edge.

In this case, $U = \max(1, k)$ ∴ we only have 2 diff values of capacities, U = k here. This will be polynomial. Also, Reduction is also polynomial time here as we are just adding some edges for n containers & m trucks $O(n+m)$.

∴ we can conclude that this is a polynomial-time algorithm.