Second Midterm

Name:

SID:

GSI and section time:

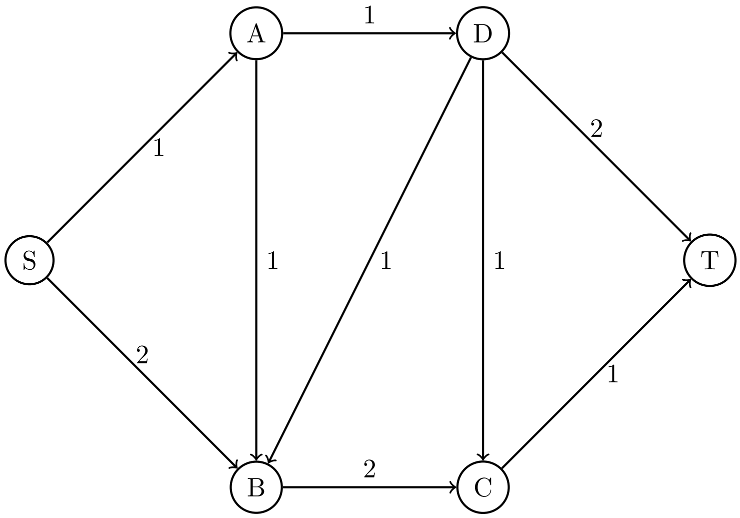
Read the questions carefully first. Be precise and concise. The number of points indicate the amount of time (in minutes) each problem is worth spending. Not all parts of a problem are weighted equally. Write in the space provided, and use the back of the page for scratch. Box numerical final answers.

Some questions are marked as "extra-credit". Only attempt these questions at the very end.

Good luck!

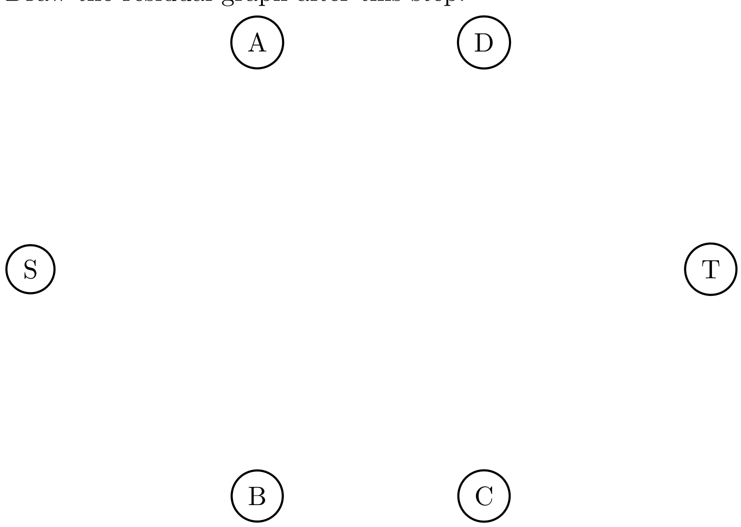
When an explanation is required, answer with one or two short sentences. *(25 points)*

1. Consider the following directed graph with all edge capacities equal to 1.



In the first step of Maximum-Flow algorithm, we increase the flow along $S \to A \to D \to B \to C \to T$ by one unit.

(a) Draw the residual graph after this step.



(b) What happens next in the execution of Max-Flow algorithm? (the algorithm does not necessarily run for three steps)

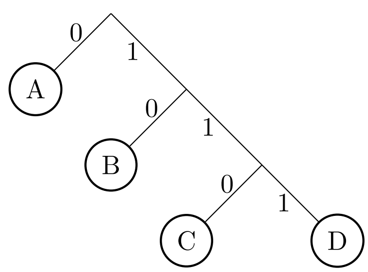
- Send unit(s) of flow on path $S \rightarrow \quad \rightarrow T$
- Send unit(s) of flow on path $S \rightarrow \quad \rightarrow T$
- Send unit(s) of flow on path $S \rightarrow \quad \rightarrow T$

(c) What is the minimum $S-T$ cut in the graph?

- S-side of the partition $= \{S, \quad \}$
- T-side of the partition $= \{T, \quad \}$

2. Give a set of frequencies on the four symbols $\{A, B, C, D\}$, for which the Huffman coding tree would be as shown below.

Symbol	Frequency
A	
B	
C	
D	



3. On a weirdly designed keyboard, every insertion takes 2 keystrokes, every deletion takes 3 key strokes and every substitution takes 4 keystrokes. Write the recurrence relation for the edit distance (minimum number of key strokes needed to edit a string $x[1, \ldots, m]$ in to a string $y[1, \ldots, n]$).

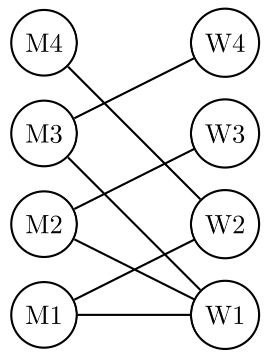
$ED[i, j] =$

Dating with Flows (5 points + 5 extra credit points)

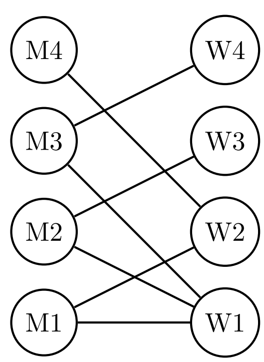
4. A dating website has used complicated algorithms to determine the compatibility between the profiles of men and women on their website. The following graph shows the set of compatible pairs.

   The website is trying to setup meetings between the men and the women. The i^{th} man has indicated a preference of meeting exactly m_i women, while the j^{th} woman prefers to meet at most w_j men. All meetings must be between compatible pairs.

   (a) How would you use the Max-flow algorithm to set up the meetings? (Draw the graph on which you would run the Max-flow) Briefly justify your answer.



   (b) *(Extra Credit, Attempt at the end, 5 points)* Suppose all the meetings need to be scheduled over 3 days and no woman wants to meet more than ℓ men on the same day. Moreover, the dating website cannot host more than m meetings in total on the same day. How would you compute the schedule using Maxflow? (Assume that the men don't mind meeting any number of women on the same day)



Covering with unit intervals (10 points)

5. Given n real numbers $x_1 < x_2 < \ldots < x_n$, we would like to cover them with the minimum number of intervals of length 1.

   For example, given $\{0.1, 0.8, 4.3, 5.1, 7.1, 7.6, 8.1\}$ we can cover it with three intervals $[0, 1]$, $[4.2, 5.2]$ and $[7.1, 8.1]$.

   (a) The greedy strategy used for the set cover problem can be applied to this problem. Give an example where this fails to find the minimum covering, and show the execution of the greedy strategy.

   (b) Describe a different greedy strategy that always finds the minimum covering. (no proof necessary)

6. True or false? Circle the right answer. No explanation needed (15 points)

(No points will be subtracted for wrong answers, so guess all you want!)

1) **T** **F** The running time of a dynamic program is at most the number of edges in its underlying DAG.

2) **T** **F** There are two ways to implement any DP algorithm: bottom-up, and via recursion with memoization. Both always have asymptotically the same time and space complexity.

3) **T** **F** Someone gives you a flow f on a graph G, claiming that f is a maximum flow. It is possible to verify this claim in $O(|E|)$ time.

4) **T** **F** Not all linear programs can be solved in polynomial time.

5) **T** **F** If a linear program has unbounded feasible region, then it does not have an optimum solution of finite value.

6) **T** **F** In succesive iterations of the Maxflow algorithm, the total flow passing through a vertex in the graph never decreases.

7) **T** **F** The running time of the algorithm for All-Pairs-Shortest-Paths would increase by a factor of 1000, if we switch the unit of measuring distances from kilometers to meters.

8) **T** **F** The running time of the algorithm for Knapsack would increase by a factor of 1000, if we switch the unit of measuring weights from kilograms to grams.

9) **T** **F** Given a Horn-SAT instance with n variables $\{x_1, \ldots, x_n\}$ and just one constraint $(x_1 \Rightarrow x_2)$, the Horn-SAT algorithm would set x_1 and x_2 to true, and all other variables to false.

10) **T** **F** If $(1,1,1)$ and $(2,2,2)$ are feasible solutions to a linear program on 3 variables then $(3,3,3)$ is also one.

11) **T** **F** If $(1,1,1)$ and $(3,3,3)$ are feasible solutions to a linear program on 3 variables then $(2,2,2)$ is also one.

12) **T** **F** The residual graph of a maximum flow f can be strongly connected.

13) **T** **F** The dynamic programming algorithm for the Travelling Salesman problem uses exponential amount of memory.

14) **T** **F** The value of edit distance between two strings of length n can be computed using $O(n)$ memory.

15) **T** **F** "Dynamic programming" sounds cool.

Water Supply via Linear Programming (15 points)

7. There are four major cities and three water reservoirs in California.

   - Reservoir i holds G_i gallons of water, while city j needs D_j gallons of water.
   - It costs p_{ij} dollars per gallon of water supplied from reservoir i to city j.
   - The capacity of the piping from reservoir i and city j can handle at most c_{ij} gallons.
   - In view of fairness, no city must get more than 1/3rd of all its water demand from any single reservoir.

   Write a linear program to determine how to supply the water from the reservoirs to the cities, at the lowest cost.

   (a) What are the variables of the linear program, and what do they indicate?

   (b) What is the objective function being maximized/minimized?

   (c) What are the constraints of your linear program? (No need to list every constraint, but list one of each type and explain how the rest are generated)

All-Pairs Shortest Path Again (15 points)

8. Here we will design a slightly slower, but intuitively simpler algorithm for All-Pairs Shortest Path in a graph G. The input is a graph $G = (V, E)$ with edge weights $c(i, j)$ between vertices i and j.

   Define the subproblem as follows:

   $$d(i, j, \ell) = \text{length of the shortest path from } i \text{ to } j \text{ that uses} < 2^{\ell} \text{ intermediate nodes.}$$

   By definition, $d(i, j, 0) = c(i, j)$.

   (a) Length of shortest path from i to j = $d(i, j, \_\_\_\_\_\_\_\_\_\_)$

   (b) Write a recurrence relation for $d(i, j, \ell)$.

   (c) Write pseudocode for computing the All-Pair-Shortest-Paths algorithm using the above recurrence.

   for $i = 1$ to n do
   for $j = 1$ to n do
   $d(i, j, 0) \leftarrow c_{ij}$

   for do

   for do

   for do

   for do

   (d) The running time of the algorithm on a graph with n vertices is ______________

Room Rentals (15 points + 15 extra-credit points)

9. You have two rooms to rent out. There are n customers interested in renting the rooms. The i^{th} customer wishes to rent one room (is happy with either room you have) for $t[i]$ days and is willing to pay $bid[i]\$$ for his/her entire stay.

   Customer requests are non-negotiable in that they would not be willing to rent for a shorter or longer duration.

   Devise a dynamic programming algorithm to determine the maximum profit that you can make from the customers over a period of D days.

   (Hint: two knapsacks?)

   (a) Briefly and precisely define the subproblems.

   (b) Write the recurrence relation between the subproblems.

(c) *(Extra Credit, Attempt at the end (15 points))* Suppose every customer has a specific start date $start[i]$ and end date $end[i]$ between which he/she is interested in renting.

How would you modify the algorithm, and the definition of your subproblems? (Give a succinct but precise description of the modifications and the subproblems. Proof of correctness & runtime analysis not needed)