Midterm II

Name:

SID:

Name and SID of student to your left:

Name and SID of student to your right:

Exam Room:

☐ Dwinelle 155	☐ Evans 10	☐ HFAX A1	☐ Dwinelle 145	☐ GPB 100	☐ Latimer 120	☐ VLSB 2040
☐ Mulford 159	☐ Kroeber 160	☐ Evans 60	☐ Soda 306	☐ Soda 320	☐ Cory 367	☐ Other

Please color the checkbox completely. Do not just tick or cross the box.

Rules and Guidelines

- **The exam is out of 135 points and will last 110 minutes.**
- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- Write your student ID number in the indicated area on each page.
- Be precise and concise. **Write in the solution box provided.** You may use the blank page on the back for scratch work, but it will not be graded. Box numerical final answers.
- The problems may **not** necessarily follow the order of increasing difficulty. ***Avoid getting stuck on a problem.***
- Any algorithm covered in lecture can be used as a blackbox. Algorithms from homework need to be accompanied by a proof or justification as specified in the problem.
- Good luck!

Discussion Section

Which of these do you consider to be your primary discussion section(s)? Feel free to choose multiple, or to select the last option if you do not attend a section. **Please color the checkbox completely. Do not just tick or cross the boxes.**

☐ Arpita, Thursday 9 - 10 am, Dwinelle 223

☐ Avni, Thursday 10 - 11 am, Dwinelle 215

☐ Emaan, Thursday 10 am - 11 pm, Etcheverry 3107

☐ Lynn, Thursday 11 am - 12 pm, Barrows 118

☐ Dee, Thursday 11 am - 12 pm, Wheeler 30

☐ Max, Thursday 12 - 1 pm, Wheeler 220

☐ Sean, Thursday 1 - 2 pm, Etcheverry 3105

☐ Neha, Thursday 2 - 3 pm, Dwinelle 242

☐ Julia, Thursday 2 - 3 pm, Etcheverry 3105

☐ Henry, Thursday 3 - 4 pm, Haviland 12

☐ Kedar, Thursday 3 - 4 pm, Barrows 104

☐ Ajay, Thursday 4-5 pm, Barrows 136

☐ Varun, Thursday 4-5 pm, Dwinelle 242

☐ Gillian, Friday 9 - 10 am, Dwinelle 79

☐ Hermish, Friday 10 - 11 am, Evans 9

☐ Vishnu, Friday 11 am - 12 pm, Wheeler 222

☐ Carlo, Friday 11 am - 12 pm, Dwinelle 109

☐ Tarun, Friday 12 - 1 pm, Hildebrand B56

☐ Noah, Friday 12 - 1 pm, Hildebrand B51

☐ Jiazheng, Friday 1 - 2 pm, Wheeler 30

☐ Claire, Friday 2 - 3 pm, Barrows 155

☐ Jialin, Friday 1 - 2 pm, Wheeler 30

☐ Teddy, Friday 1 - 2 pm, Dwinelle 105

☐ Jierui, Friday 2 - 3 pm, Wheeler 202

☐ David, Friday 2 - 3 pm, Wheeler 130

☐ Nate, Friday 2 - 3 pm, Evans 9

☐ Rishi, Friday 3 - 4 pm, Dwinelle 243

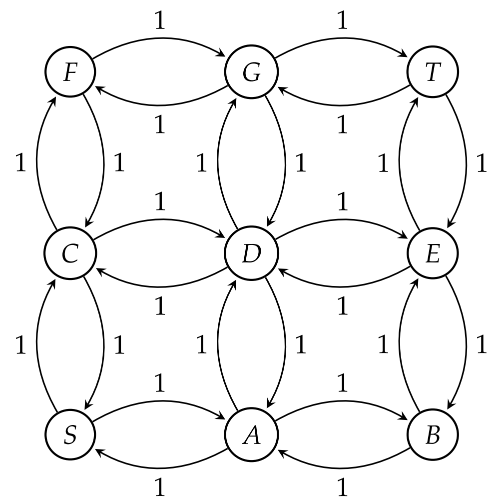
☐ Tiffany, Friday 3 - 4 pm, Barrows 104

☐ Ida, Friday 3-4 pm, Dwinelle 109

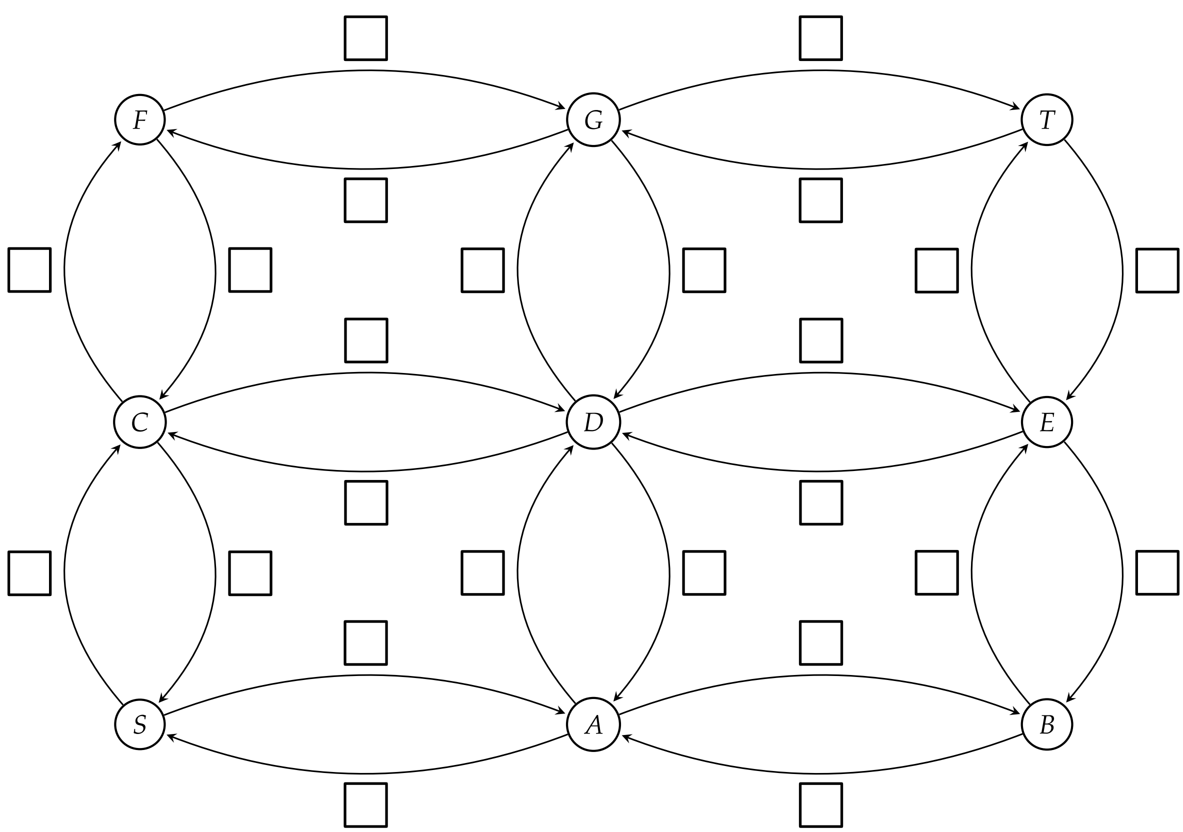
☐ Don't attend section.

1 Max Flow

Recall that the Ford-Fulkerson algorithm iteratively uses the residual network to compute the max flow. Consider the following network:



(i) (**4 points**) Let f be a flow that assigns 1 unit of flow on the path $S \to A \to B \to E \to D \to C \to F \to G \to T$ and 0 flow elsewhere. Write down the residual capacities of the edges after routing the flow f.



(ii) (**2 points**) What path does the algorithm use in the next iteration?

(iii) (**3 points**) Write down the total incoming flow to all these vertices in the graph after these two iterations.

Node	Total Incoming Flow
A	
B	
C	
D	
E	
F	
G	

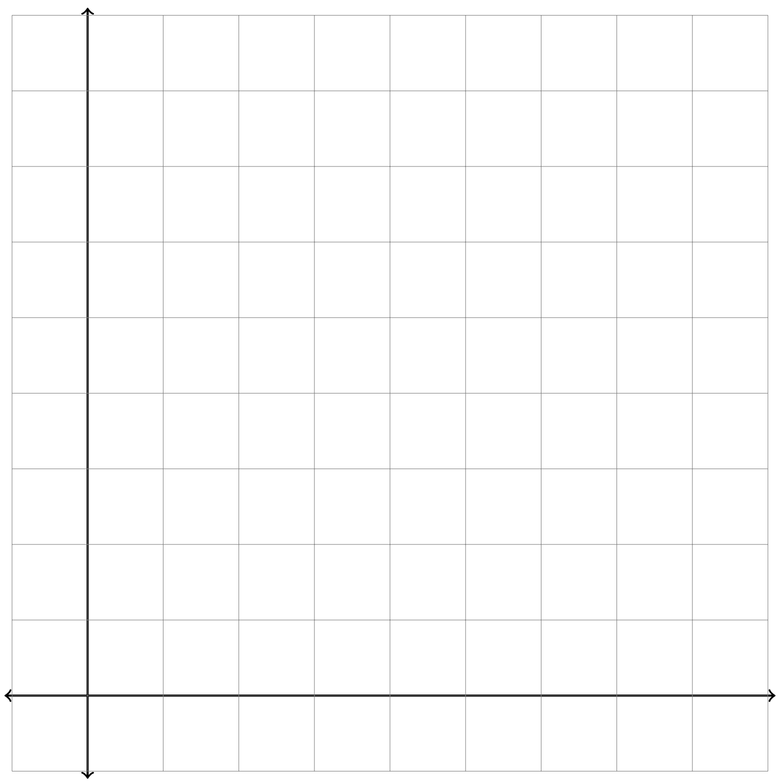
(iv) (**1 point**) What is the value of the maximum flow on this network?

2 Linear Programming

Consider the following linear program.

$$
\begin{aligned}
\text{Maximize } & x + 2y \\
\text{Subject to } & -3x + y \leq 1 \\
& y - x \leq 3 \\
& y - x \geq -3 \\
& x, y \geq 0
\end{aligned}
$$

(i) (**6 points**) Draw out the feasible region for the LP.



(ii) (**3 points**) Write the dual linear program (use variables a, b, c).

(iii) (**2 points**) The number of vertices of the feasible region of the dual linear program is

3 Dynamic Programming: Alternate Subproblems

Choosing the right sub-problems is critical in the design of dynamic programming algorithms. Here we will consider alternate subproblems for DP algorithms we studied in the class.

In each case, specify whether it is possible to write a recurrence relation for the subproblem defined. If Yes, then write the correct recurrence relation (Base case not needed). If you believe that it is not possible, select No (no justification needed).

You may not use any external datastructures/alternate subproblems, only the ones provided.

The scoring on the question is as follows:

- Correctly answer Yes, and the correct recurrence relation: 5 points (partial credit for recurrence relation, but no credit for just "Yes").
- Correctly answer No: 3 points
- Incorrectly answer No: -3 points
- Leave it blank: 0 points

(i) **Longest increasing subsequence in the array** $a[1], \ldots, a[n]$
Subproblem:

$$LIS[i] = \text{ length of longest increasing subsequence contained within } a[1], \ldots, a[i],$$

for all $i \in \{1, \ldots, n\}$

○ Yes, it is possible ○ No, there is no recurrence relation

(ii) **Knapsack with repetition with** n **items with values** v_i **and weight** w_i**.**
Subproblem:

$$K[v] = \text{ weight of the lightest collection with total value at least } v \text{ (repetitions allowed)}$$

for all $v \in \{1, \ldots, V\}$

○ Yes, it is possible ○ No, there is no recurrence relation

(iii) **Edit Distance between strings $x[1,\ldots,n]$ and $y[1,\ldots,m]$.**
Subproblems:

$$ED[i,j] = \text{edit distance between the prefix } x[1,\ldots,i] \text{ and suffix } y[j\ldots m]$$

for all $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,m\}$

○ Yes, it is possible ○ No, there is no recurrence relation

(iv) **Knapsack without repetition with n items with values v_i and weight w_i.**
Subproblems:

$$K[w,i,1] = \text{ maximum value of a collection of items with total weight } w \text{ that contains item } i$$
$$K[w,i,0] = \text{ maximum value of a collection of items with total weight } w \text{ that does not contain item } i$$

for all $w \in \{1,\ldots,W\}$ and $i \in \{1,\ldots,n\}$

○ Yes, it is possible ○ No, there is no recurrence relation

(v) **All-pairs shortest paths in a graph** $G = (V, E)$ **with edge weights** $\{w_e\}_{e \in E}$**.**

Subproblem:

$d[i, j, k] =$ length of shortest path from i to j among those that do not pass through $\{1, \ldots, k\}$ in the intermediate steps.

for all $i, j, k \in \{1, \ldots, n\}$.

○ Yes, it is possible ○ No, there is no recurrence relation

4 LP again

(i) (**3 points**) How many **vertices** does the following feasible region have?

$$-1 \leq x_1 \leq 1$$
$$-1 \leq x_2 \leq 1$$
$$-1 \leq x_3 \leq 1$$
$$-1 \leq x_4 \leq 1$$
$$-1 \leq x_5 \leq 1$$

(ii) (**5 points**) Consider the following linear program:

$$x + 2y + 3z \leq 6 \quad (1)$$
$$2x + 5y + 7z \leq 14 \quad (2)$$
$$3x + y + 5z \leq 9 \quad (3)$$
$$x + 4y \leq 12 \quad (4)$$
$$x \geq 0 \quad (5)$$
$$y \geq 0 \quad (6)$$
$$z \geq 0 \quad (7)$$

Two of these points are NOT vertices of the feasible region, mark them.

(a) $(3,0,0)$

○ Not a vertex

(b) $(1,1,1)$.

○ Not a vertex

(c) $(0,2,0)$

○ Not a vertex

(d) $(2,2,0)$

○ Not a vertex

(e) $(0,0,1)$

○ Not a vertex

(iii) (**2 points**) Suppose $(x,y) = (3,-1)$ and $(x,y) = (-1,3)$ are both feasible solutions to some linear program. Write down another point that is definitely a feasible solution to the same linear program.

(iv) (**2 points**) Consider primal(left) and dual(right) LP:

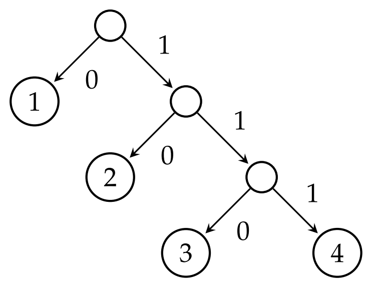
$$\begin{array}{cc} \max \mathbf{c}^T\mathbf{x} & \min \mathbf{y}^T\mathbf{b} \\ \mathbf{A}\mathbf{x} \leq \mathbf{b} & \mathbf{y}^T\mathbf{A} \geq \mathbf{c}^T \\ \mathbf{x} \geq 0 & \mathbf{y} \geq 0 \end{array}$$

(a) If we change the primal objective to $\max 2c^T x$, what is the new dual objective function?

(b) Let D^* be the value of the optimal solution to the dual. If we change the primal objective to $\max 2c^T x$, what would value of the optimal solution to the new dual be (in terms of D^*)?

5 Greedy

(i) (**5 points**) Consider the following Huffman tree (denote T) shown below:



Let $\mathcal{F}$ denote the set of all frequency vectors $f = (f_1, f_2, f_3, f_4)$ for which the above tree is the optimal Huffman tree, i.e., Formally,

$$\mathcal{F} = \{(f_1, f_2, f_3, f_4) \in \mathbb{R}^4 | T \text{ is an optimal Huffman tree for frequencies } (f_1, f_2, f_3, f_4)\} \tag{8}$$

Write an LP whose feasible region is $\mathcal{F}$.

(ii) (**4 points**) The greedy algorithm for Minimum Set Cover is executed on an input consisting of a family of subsets $\{S_1, \ldots, S_m\}$ where each $S_i \subseteq \{1, \ldots, 1000\}$.

(a) In the first iteration, the greedy algorithm picks a set of size 200 (thereby covering 200 elements). What is the smallest possible value of the optimal solution?

(b) In the second iteration, the greedy algorithm picks a set of size 100 that covers 10 additional elements. What is the smallest possible value of the optimal solution?

6 Zero Sum Games (10 points)

Consider a zero-sum game given by a $n \times n$ payoff matrix P. Let $\mathcal{R}_{pure}$ and $\mathcal{R}_{mixed}$ denote the sets of pure and mixed strategies for the row-player. Similarly, let $\mathcal{C}_{pure}$ and $\mathcal{C}_{mixed}$ denote the set of pure and mixed strategies for the column player.

For a row-player strategy r and a column player strategy c, let $\mathsf{P}(r, c)$ denote the payoff of the row player when the players use strategies r and c. Depending on which player goes first and what strategies they can use, one can define several different quantities associated with the game. In each of the following cases, write down whether the LHS is $\geq, =, \leq$ than the RHS. (Write the strongest identity that is true)

(i)

$$\min_{c \in \mathcal{C}_{pure}} \left(\max_{r \in \mathcal{R}_{pure}} \mathsf{P}(r, c) \right) \qquad \square \qquad \max_{r \in \mathcal{R}_{pure}} \left(\min_{c \in \mathcal{C}_{pure}} \mathsf{P}(r, c) \right)$$

(ii)

$$\min_{c \in \mathcal{C}_{mixed}} \left(\max_{r \in \mathcal{R}_{mixed}} \mathsf{P}(r, c) \right) \qquad \square \qquad \max_{r \in \mathcal{R}_{mixed}} \left(\min_{c \in \mathcal{C}_{mixed}} \mathsf{P}(r, c) \right)$$

(iii)

$$\min_{c \in \mathcal{C}_{pure}} \left(\max_{r \in \mathcal{R}_{mixed}} \mathsf{P}(r, c) \right) \qquad \square \qquad \min_{c \in \mathcal{C}_{mixed}} \left(\max_{r \in \mathcal{R}_{mixed}} \mathsf{P}(r, c) \right)$$

(iv)

$$\min_{c \in \mathcal{C}_{pure}} \left(\max_{r \in \mathcal{R}_{mixed}} \mathsf{P}(r, c) \right) \qquad \square \qquad \min_{c \in \mathcal{C}_{pure}} \left(\max_{r \in \mathcal{R}_{pure}} \mathsf{P}(r, c) \right)$$

7 Multiplicative Weights (True/False) (8 points)

Let $w_1^{(t)}, \ldots, w_n^{(t)}$ be the weights of the multiplicative weights update algorithm after t steps. If $w_i^{(t)} > w_j^{(t)}$ then which of the following statements is true.

(i) Expert i had a smaller total loss than expert j on the first t steps.

◯ True ◯ False

(ii) Expert i is more likely to have a smaller loss than Expert j in the step $(t+1)$.

◯ True ◯ False

(iii) Expert i is more likely to have a smaller loss than Expert j in the step $(t+1)$, only if the step t is sufficiently large.

◯ True ◯ False

(iv) Multiplicative weights algorithm is more likely to pick expert i than expert j in the step $t+1$.

◯ True ◯ False

8 True/False (10 points)

(i) Permuting the order of the clauses in a Horn-SAT formula can change the solution produced by the greedy algorithm for it.

◯ True ◯ False

(ii) Suppose that decreasing the capacity of edge $u \to v$ decreases the maximum flow from s to t in a network G. Then u and v must be on different sides of every minimum cut of the graph G.

◯ True ◯ False

(iii) Suppose that increasing the capacity of edge $u \to v$ increases the maximum flow from s to t in a network G. Then u and v must be different sides of every minimum cut of the graph.

◯ True ◯ False

(iv) Simplex runs in polynomial time $O(n^c)$ for some fixed $c \in \mathbb{N}$, over all inputs of size n.

◯ True ◯ False

(v) Suppose that $S_1, S_2 \subset \mathbb{R}^n$ denote the feasible regions of two linear programs LP_1 and LP_2, then $S_1 \setminus S_2$ is also a feasible region of a different LP. (Recall that for two sets A and B, $A \setminus B$ denotes those elements that are in A but not in B).

◯ True ◯ False

9 Fault-Tolerant Road Network (16 points)

There are n cities in the country Pessimia and you are given positive distances $\{d_{ij}\}_{i,j\in\{1,\ldots,n\}}$ between them.

The government of Pessimia would like to build a road network $G = (\{1,\ldots,n\}, E)$ between the cities. The government would like to build as many roads between cities as possible. But the road network G needs to be fault-tolerant in the following sense:

If any road $(i,j) \in E$ is deleted from G, then the network G still contains an alternate path from i to j of length at most $2d_{ij}$.

1. Describe an algorithm to find the fault-tolerant road network $G = (V, E)$ with the maximum number of roads. Your algorithm can use shortest path algorithms as a black box, and would need to run in time at most $O(|V|^6 + |E|^3)$.

   (Hint: Greedy)

2. Briefly justify the correctness of your algorithm.

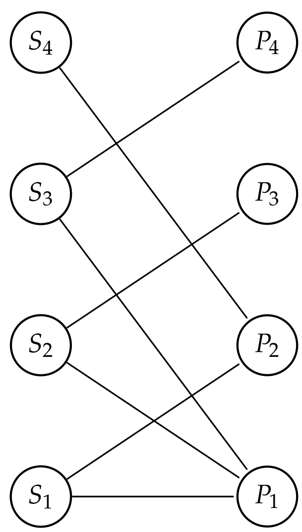
10 Matching Pets (12 points)

An animal shelter has profiles of students and pets in the shelter, as well as a list of compatible pairs among students and pets. The shelter is trying to setup meetings between students and the pets in the shelter.

1. (**4 points**) A compatibility graph between 4 students and 4 pets is shown below. Here are the constraints on the meetings:

   - The i^{th} student S_i has indicated a preference of meeting exactly s_i pets.
   - The j^{th} pet P_j can meet at most p_j students.
   - All meetings must be between compatible pairs.

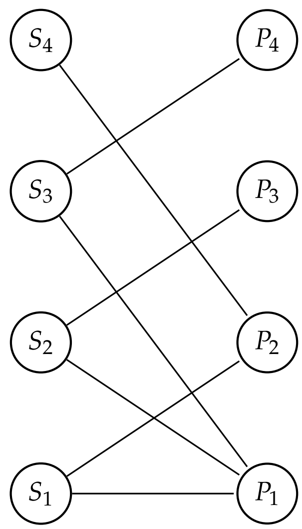
   You would like to use one Max-Flow computation to decide whether one can setup the meetings so that the preferences of all students and pets are satisfied. Starting with the compatibilities provided, draw the network on which one would run the Max-flow algorithm. Indicate the source of the flow by S, the sink by T and mark the capacities of all edges.



2. (**8 points**) The compatibility graph between 4 students and pets is shown below. Here are the constraints on the meetings:

   - The meetings need to be scheduled over 3 days.
   - The animal shelter cannot host more than m meetings in total on the same day.
   - No pets wants to meet more than ℓ students on the same day, but the students don't mind meeting any number of pets on the same day.
   - The i^{th} student S_i has indicated a preference of meeting exactly s_i pets.
   - The j^{th} pet P_j can meet at most p_j students.
   - All meetings must be between compatible pairs.

   You would like to use a single Max-Flow computation to decide which pairs of students and pets should meet on each day. Starting with the compatibilities provided, draw the graph on which one would run the Max-flow algorithm. Indicate the source by S, the sink by T and mark the capacities of all edges.



11 Maximum Connected Subgraph (18 points)

You are given a tree $T = (V, E)$ with vertex weights $w(v)$. The weights can be any real number (positive or negative). You wish to find the largest possible weight of any connected subgraph $S \subseteq V$, where the weight of S is defined as $\sum_{s \in S} w(s)$. Note that if S is empty, that is a valid subgraph with total weight 0.

(a) (**4 points**) In *maximum contiguous subarray*, you are given an array $[a_1, \ldots, a_n]$ where a_n are arbitrary real numbers, and you want to find the maximum-sum contiguous subarray. Describe how to use an algorithm for maximum connected subgraph as a black-box to solve maximum contiguous subarray.

(b) (**4 points**) Henry fixes an arbitrary vertex r to be the root of the tree, and proposes the following dynamic programming algorithm to find the weight of maximum connected subgraph:

- *Subproblems*: $N[v]$ - the weight of the largest connected subgraph in the subtree of v
- *Base cases*: if v is a leaf, $N[v] = \max(0, w(v))$
- *Recurrence*: $N[v] = \max(0, w(v)) + \sum\{N[c] \mid c \text{ child of } v, N[c] > 0\}$
- *Return*: $N[r]$ where r is the root

Give an example of a tree on 3 vertices along with weights for which Henry's algorithm fails.

(c) (**5 points**) Now, design a dynamic programming algorithm to *correctly* find the weight of the maximum-weight connected subgraph. First, define your subproblem(s).

(d) (**5 points**) Write the recurrence relation for the subproblem(s).