Midterm 2

Name:

SID:

Name and SID of student to your left:

Name and SID of student to your right:

Circle One: Pimentel Wheeler Wozniak

Rules and Guidelines

- **The exam is out of 80 points and will last 80 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.
- Any algorithm covered in lecture can be used as a blackbox, unless otherwise stated.
- If no specific runtime is provided in the question, find as efficient an algorithm as you can. If a runtime is specified and you cannot achieve it, but you can get a weaker runtime, then submit that solution.
- Good luck!

Discussion Section [1 point]

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.**

☐ Aarash, Tuesday 5 - 6 pm, Barrows 151

☐ Nick T., Tuesday 5 - 6 pm, Wheeler 202

☐ Chinmay, Wednesday 9 - 10 am, Dwinelle 215

☐ Chinmay, Wednesday 10 - 11 am, Moffitt 150D

☐ Simin, Wednesday 10 - 11 am, Mulford 240

☐ Aditya M., Wednesday 11 am - 12 pm, Barrows 56

☐ Nick W., Wednesday 11 am - 12 pm, Giannini 141

☐ Yuxiang, Wednesday 1 - 2 pm, Dwinelle 215

☐ Nikhil, Wednesday 1 - 2 pm, Wheeler 108

☐ James, Wednesday 1 - 2 pm, Soda 405

☐ Aditya B., Wednesday 2 - 3 pm, Wheeler 200

☐ Owen, Wednesday 2 - 3 pm, Etcheverry 3105

☐ James, Wednesday 2 - 3 pm, Soda 310

☐ Aditya B., Wednesday 3 - 4 pm, Wheeler 202

☐ Harley, Wednesday 3 - 4 pm, Wheeler 220

☐ Michael, Wednesday 3 - 4 pm, Soda 310

☐ Vinay, Wednesday 4 - 5 pm, Dwinelle 223

☐ Benjamin, Wednesday 5 - 6 pm, GPB 107

☐ Mudit, Wednesday 5 - 6 pm, Moffitt 150D

☐ I do not attend a discussion section

1 Multiple Choice [2 points per problem]

Fill in a single square for each problem. Fill it in completely.

(a) Say we run the Huffman encoding algorithm on a file with n *distinct* symbols, each of which appears an equal number of times. What are the lengths of the longest and shortest resulting codewords? (Assume that n is a power of 2)

- ☐ Longest $n-1$; shortest 1
- ☐ Longest $\log(n)$; shortest $\log(n)$
- ☐ Longest $\frac{n}{2}$; shortest $\frac{n}{2}$
- ☐ Longest n; shortest $n-1$

(b) Say we had a similar setup to the last part, except that symbol i appears 2^i times ($0 \leq i \leq n-1$). Now what are the lengths of the longest and shortest codewords? (Assume that n is a power of 2)

- ☐ Longest $n-1$; shortest 1
- ☐ Longest $\log(n)$; shortest $\log(n)$
- ☐ Longest $\frac{n}{2}$; shortest $\frac{n}{2}$
- ☐ Longest n; shortest $n-1$

(c) Suppose we have a connected graph G on n vertices such that three of the edges in G have weight 0 while all other edges have distinct positive weights (ie, other than the original three, no pair of edges can have the same weight). What is the maximum number of MSTs G could have? What is the minimum number?

- ☐ Maximum 1; minimum 1
- ☐ Maximum $\lfloor \frac{n}{3} \rfloor$; minimum 1
- ☐ Maximum 3; minimum 1
- ☐ Maximum 3; minimum 3

(d) Consider the following Horn SAT formula on five variables:

$$(c \wedge e) \Rightarrow a,\ (a \wedge c) \Rightarrow d,\ \Rightarrow a,\ e \Rightarrow c,\ a \Rightarrow b,$$
$$(\overline{a} \vee \overline{d} \vee \overline{e})$$

How many satisfying assignments are there for this formula?

☐ 0

☐ 1

☐ 2

☐ 3

(e) Suppose we have some instance of the Knapsack problem. Let U be the value of the optimal solution if there is an unlimited number of each item available and V be the optimal value if there is only one of each item available. What is the strongest relationship we can give between U and V?

- ☐ $U \leq V$
- ☐ $U \geq V$
- ☐ $U = V$
- ☐ None of the above are guaranteed to be true.

For the next 2 parts, let $G = (V, E, w)$ be an undirected weighted graph with *distinct* edge weights w. Let $G' = (V, E, w')$ be the same graph with weights defined by $w'(e) = w(e) + 1$.

(f) Suppose you ran Prim's Algorithm on G and G'. Would the spanning trees produced be the same?

- ☐ Always
- ☐ Never
- ☐ Depends on the graph G

(g) Now suppose you ran Dijkstra's Algorithm on G and G' for vertices $s, t \in V$. Would the shortest path produced be the same?

- ☐ Always
- ☐ Never
- ☐ Depends on the graph G

2 "The Final Problem" [10 points]

Sherlock Holmes boards the train from London to Dover in an effort to reach the continent and so escape from Professor Moriarty. Moriarty can take an express train and catch Holmes at Dover. However, there is an intermediate station at Canterbury at which Holmes may detrain to avoid such a disaster. But of course, Moriarty is aware of this too and may himself stop instead at Canterbury. Von Neumann and Morgenstern once estimated the value to Moriarty of these four possibilities to be given in the following matrix (in some unspecified units).

		Holmes	
		Canterbury	Dover
Moriarty	Canterbury	200	-100
	Dover	0	200

Assume Moriarty goes to Canterbury with probability x_1 and Dover with probability x_2 for x_1, x_2 known to Holmes. What is the expected payoff to Moriarty assuming Sherlock plays optimally? Formulate this as a minimization linear program. You do not need to express the linear program in canonical form.

$$\text{OPT}_{\text{row}} = \begin{cases} \max \\ \\ \text{s.t.} \\ \\ \\ \\ \end{cases}$$

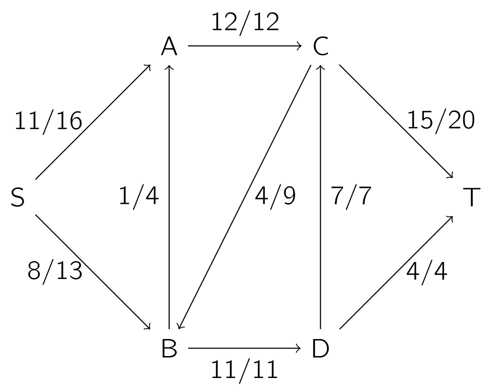
What are the optimal strategies for Holmes and Moriarty, and what is the value of the game? (Historically, as related by Dr. Watson in *The Final Problem* in Arthur Conan Doyle's *The Memoires of Sherlock Holmes*, Holmes detrained at Canterbury and Moriarty went on to Dover.)

Answer:

3 Computing Flows [15 points]

Suppose you run Ford-Fulkerson on the given graph to compute the max-flow from vertex A to vertex F. After several iterations you end up with following flow (an edge labelled "x/y" denotes an edge of capacity y with x flow going through it):

Draw the residual graph for this flow.



A C

S T

B D

What is an augmenting path that the Ford-Fulkerson algorithm could pick on its next iteration. How much flow would be sent along this path?

Draw the new residual graph after executing the iteration described in the previous part.

A C

S T

B D

Will Ford-Fulkerson execute for another iteration? If so, detail another path it could pick and the flow along it. If not, state the min-cut of the graph.

4 Duality [10 points]

Fill in the boxes for the dual ($\mathcal{D}$) to the following linear program ($\mathcal{P}$).

$$(\mathcal{P}) = \begin{cases} \max & 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 5 \\ & 3x_1 + 4x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{cases} \qquad (\mathcal{D}) = \begin{cases} \min & \boxed{}\, y_1 + \boxed{}\, y_2 \\ \text{s.t.} & \boxed{}\, y_1 + \boxed{}\, y_2 \geq \boxed{} \\ & \boxed{}\, y_1 + \boxed{}\, y_2 \geq \boxed{} \\ & y_1, y_2 \geq 0 \end{cases}$$

Show that *any* feasible solution of the dual provides an upper bound to *any* feasible solution of the primal. You may either write a couple of sentences arguing how the construction of the dual program necessarily implies this, or actually work through the math to prove it.

5 Fixed leaf MST [15 points]

Given an *undirected* weighted graph $G = (V, E)$ and a subset $S \subseteq V$ of the vertices, devise an algorithm that generates a spanning tree of minimum weight such that each vertex $s \in S$ is a leaf of the tree. Note that the tree is allowed to have additional leaves not in S. Recall that a leaf of a tree is a vertex with degree 1. If it is not possible to construct such a spanning tree, then your algorithm should return "NOT POSSIBLE" or some similar indicator. For partial credit, solve the case when $|S| = 1$.

Provide just your main idea and a brief justification. For full credit your algorithm should be as efficient as possible.

Solution:

6 Successful passage through Gridworld [15 points]

You have been flattened into a 2D occupant of Gridworld, which is a $n \times n$ sized grid. Your goal is to move from $(0, 0)$ to (n, n). You are given a map of Gridworld, the matrix $G \in \mathbb{Z}^{n \times n}$, and know that while traveling *out* of any given cell, you will either pay tokens for passage or be gifted tokens by a generous stranger. The value you will lose/gain at (i, j) is recorded in $G(i, j)$; $G(i, j)$ is positive if you gain tokens. You can only continue moving on the grid when you have ≥ 0 tokens, and you may move from (i, j) to $(i + 1, j)$ or $(i, j + 1)$.
Give a dynamic programming algorithm to compute the minimum number of tokens you need to begin with (at (0,0)) to ensure successful passage.

Define your subproblem(s):

What are the base cases?

Write the recurrence relation for the subproblems.

What is the time complexity of your algorithm in Big Theta notation? What is the best possible space complexity you can achieve? Briefly explain.

Blank page: If using this page to write an answer, clearly specify on the question that your work should be found here.

Blank page: If using this page to write an answer, clearly specify on the question that your work should be found here.