CS 4050 Test 2 Name Matt Hart

Each problem is worth 8 test points (purely for assigning a score out of 100% on the test), and some problems may have several unrelated parts worth the points indicated.

Please write your answers on these sheets or your own paper and send me either one PDF file for your entire test or a separate file for each problem.

You may use any of our course materials: the written chapter documents, the videos, and your Exercise work. Other than accessing those allowed materials, you *may not* run any programs, including any Web browser or communication software, to help you on any questions, or communicate with anyone other than me about the test questions or course material. Except, of course, one problem explicitly says that you should use your computer to run `ManualSimplex`, so just on that problem you are allowed to use a computer.

---

1. Show all the details to efficiently compute $12P$ for elliptic curve cryptography, using $p = 13$, using the curve $y^2 = x^3 + 5x + 2$ (i.e., $a = 5$ and $b = 2$), and using the starting point $P = (11, 7)$.

You will be penalized if you do an inefficient algorithm,

To save some routine work, you should rely on these values of mP:

$$1P = (11, 7)$$

$$2P = (7, 9)$$

$$4P = (9, 3)$$

For your convenience, here is the multiplication table for Z_{13}:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	2	4	6	8	10	12	1	3	5	7	9	11
3	0	3	6	9	12	2	5	8	11	1	4	7	10
4	0	4	8	12	3	7	11	2	6	10	1	5	9
5	0	5	10	2	7	12	4	9	1	6	11	3	8
6	0	6	12	5	11	4	10	3	9	2	8	1	7
7	0	7	1	8	2	9	3	10	4	11	5	12	6
8	0	8	3	11	6	1	9	4	12	7	2	10	5
9	0	9	5	1	10	6	2	11	7	3	12	8	4
10	0	10	7	4	1	11	8	5	2	12	9	6	3
11	0	11	9	7	5	3	1	12	10	8	6	4	2
12	0	12	11	10	9	8	7	6	5	4	3	2	1

$y^2 = x^3 + 5x + 2$

$y = \sqrt{x^3 + 5x + 2}$

EC equation

$y^2 = x^3 + 5x + 2 \pmod{p}$

You may use a calculator to do arithmetic, including reducing things mod 13, but what you write down for your answer must clearly explain all the steps that are necessary to compute the desired result.

$4P = 2(7,9)$ $\quad \lambda = \dfrac{3x_p^2 + a}{2y_p} \Rightarrow \dfrac{3\cdot 7^2 + 5}{2\cdot 9} = \dfrac{152}{18} = \dfrac{76}{9} = \boxed{7}$

$\boxed{\mathbb{Z}_{13}\ \dfrac{1}{9} = 3}$ $\quad \boxed{3 \% 13 = 7}$

$\boxed{\lambda = 7}$

$x_R = \lambda^2 - x_p - x_p \Rightarrow 7^2 - 7 - 7 = 35 \% 13 = \boxed{9}$

$4P(x_R, y_R)$

$4P(9, 3)$ ✓

$y_R = y_p + \lambda(x_R - x_p) = 9 + 7(9 - 7)$

$= 23 \% 13$

$-R = 10 \% 13 = \boxed{3}$

$8P = 2(9,3)$

$$\frac{3x_p^2 + a}{2y_p} \Rightarrow \frac{3 \cdot 9^2 + 5}{2 \cdot 3} = \frac{248}{6} = \frac{124}{3} \quad \frac{1}{3}\ Z_{13} = \textcircled{9}$$

$8P = (x_R, y_R)$

$$\lambda = 9 \cdot 124 \ \% \ 13 = \textcircled{11}$$

$\boxed{8P = (12, 3)}$

$$x_R = \lambda^2 - x_p - x_p \Rightarrow 11^2 - 9 - 9 = 103 \ \% \ 13 = \textcircled{12}$$

$$y_R = y_p + \lambda(x_R - x_p) = 3 + 11(12 - 9) = 36 \% 13 = 10 \ \% \ 13 = 3$$

$$-R = -10 \ \% 13 = \textcircled{3}$$

$8P + 4P = 12P$

$$(12,3) + (9,3) \Rightarrow \lambda = \frac{y_2 - y_1}{x_2 - x_1} \Rightarrow \lambda = \frac{3 - 3}{9 - 12} = \frac{\emptyset}{-5} = \emptyset$$

$\boxed{\lambda = 0}$

y_1 x_1 x_2

$$x_R = \lambda^2 - x_p - x_Q = 0^2 - 12 - 9 = -21 \% 13 = \textcircled{5}$$

$$y_R = y_p + \lambda(x_R - x_p) = 3 + 0(5 - 12) = -5 \ \% \ 13 = \textcircled{10}$$

$12P \ \boxed{R = (5, 10)}$

$-R = (5, -3)$

doublings		halvings	Accumulated answer	
P	$(11,7)$	12		even, dont + P
$2P$	$(7,9)$	6		even, dont + 2P
$4P$	$(9,3)$	3	$P + 2P$	odd, add 4P
$8P$	$(12,3)$	1	$P + 4P$	odd, add 8P
$12P$	$(5,10)$		$4P + 8P = 12P$	

Name Matt Hurt

2. Consider the instance of the 0-1 knapsack problem with capacity 36 and these items:

j	p_j	w_j
1:	414	18
2:	252	12
3:	140	7
4:	162	9
5:	128	8
6:	45	3
7:	24	2
8:	40	4

$w_j = 36$ ↓

$P_j = \max P$ ↓

18, 12, 2 = \$730

1,3,4,7 = \$740

12,7,9,8 = \$702

Here is the last 4 rows of the dynamic programming table (the first rows and smaller columns have been omitted), where as usual the column label is the weight allowed for the cell, the upper number in the cell is the maximum profit achievable with items 1 through 5 allowed with the given weight, and the lower quantity is the set of items that achieves that profit:

wt		24	25	26	27	28	29	30	31	32	33	34	35	36
8	5	430 24 ~~{3, 4, 5}~~	554 25 ~~{1, 3}~~	554 25 ~~{1, 3}~~	576 27 ~~{1, 4}~~	576 27 ~~{1, 4}~~	576 27 ~~{1, 4}~~	666 30 ~~{1, 2}~~	666 30 ~~{1, 2}~~	666 30 ~~{1, 2}~~	682 33 ~~{1, 3, 5}~~	716 34 ~~{1, 3, 4}~~	716 34 ~~{1, 3, 4}~~	716 34 ~~{1, 3, 4}~~
3	6	45+430 475 {3,4,5,6}			621 30 {1,4,6}			666 30 {1,2}		666 30 {1,2}		716 34 {1,3,4}		727 36 {1,3,5,6}
2	7									690 32 {1,2,7}				740 36 {1,3,4,7}
4	8													740 36 {1,3,4,7}

or we could exceed the wt cost capacity to 37 and collect the profit of 761

Your job on this problem is to complete the solution of this instance, filling in exactly the cells that are needed, and marking somehow the cells in row 5 that are needed. For each cell that you fill, be sure to give its optimal profit and the set of items that achieve that profit. Finally, state clearly the final solution to the instance: what items should be selected, what do they weigh in total, and what profit do they give?

P{1,3,4,7}= 36

Max profit = \$740

Name Matt Hunt

3. Below you will find the matrix $D^{(3)}$ obtained part of the way through doing Floyd's algorithm on a given set of edge weights.

Do the next steps of Floyd's algorithm from this point, computing what cells are improved by allowing vertex 4 to be used as an intermediate vertex, and then show what cells are improved by allowing vertex 5 to be used as an intermediate vertex.

Mark any changes directly in the given diagram, crossing out old values and writing in new ones.

Be sure to update both numbers in any cell where a change is made.

	1	2	3	4	5
1	0 0	1 ✓ 0 ✓	3 ✓ ~~2~~ 0	6 ✓ ~~3~~ 0	~~∞~~ 10 ✓ 4 ~~0~~
2	8 ✓ 0 ✓	0 0	2 ✓ 0 ✓	5 ✓ ~~3~~ 0	~~∞~~ 9 ✓ 4 ~~0~~
3	15 ✓ ~~2~~ 0	7 0	0 0	3 0	~~∞~~ 7 4 ~~0~~
4	21 ✓ ~~3~~ 0	13 ✓ ~~3~~ 0	6 0	0 0	4 ✓ 0 ✓
5	~~∞~~ 26 ✓ 4 ~~0~~ ✓	~~∞~~ 18 ✓ 4 ~~0~~ ✓	11 ~~∞~~ 4 ~~0~~	5 ✓ 0 ✓	0 0

$D^{(1)}$

$D^{(2)}$

$D^{(3)}$

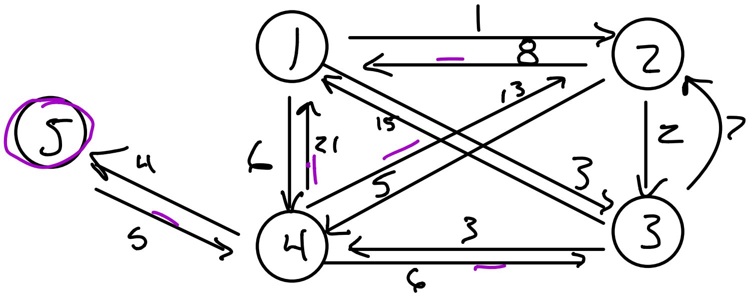
$D^{(4)}$

$D^{(5)}$

$5 \xrightarrow{5} 4 \xrightarrow{21} 1$

$5 + 21 = 26$

Then show how to use the chart to find the optimal path from vertex 5 to vertex 1, assuming that any of vertices 1 through 5 can be used as intermediate vertices. Be sure to show the entire sequence of vertices with the cost of each edge, and to verify that the path cost is the same as in the chart.



4. Here are some of the weights for a directed graph and a chart for the dynamic programming approach to the Traveling Salesman Problem on the given graph:

	1	2	3	4	5
1	0	4	7	6	5
2	2	0	3	2	4
3	2	6	0	8	1
4	3	5	1	0	2
5	6	16	4	12	0

Your job on this problem is to clearly show how this dynamic programming information can be used to find an optimal tour for the given graph. Circle all chart items that you use, show all your computations, and clearly state the vertices of the optimal tour in order and the total weight of the optimal tour.

```
                   2          3          4          5
         {}  2.00( 0)  2.00( 0)  3.00( 0)  6.00( 0)
        {2} ---------  8.00( 2)  7.00( 2) 12.00( 2)
        {3}  5.00( 3) ---------  3.00( 3)  6.00( 3)
      {2,3} --------- ---------  9.00( 3) 12.00( 3)
        {4}  5.00( 4) 11.00( 4) --------- 15.00( 4)
      {2,4} --------- 11.00( 2) --------- 15.00( 2)
      {3,4}  5.00( 4) --------- --------- 15.00( 3)
    {2,3,4} --------- --------- --------- 15.00( 2)
        {5} 10.00( 5)  7.00( 5)  8.00( 5) ---------
      {2,5} --------- 13.00( 5) 14.00( 5) ---------
      {3,5} 10.00( 3) ---------  8.00( 3) ---------
    {2,3,5} --------- --------- 14.00( 3) ---------
      {4,5} 10.00( 4) 16.00( 4) --------- ---------
    {2,4,5} --------- 16.00( 2) --------- ---------
    {3,4,5} 10.00( 4) --------- --------- ---------
  {2,3,4,5} --------- --------- --------- ---------
```

1 → 2 = 4
2 → 4 = 2
4 → 3 = 1   tour
3 → 5 = 1
5 → 1 = 6

Total weight 14 best!

1 → D[{2,4,5}][3] = 23 (3)
7 + 16 = 23 ✓

1 → D[{2,3,5}][3] = 20 (4)
6 + 14 = 20 ✓

1 → D[{2,3,4}][5] = 20 (5)
5 + 15(2) = 20 ✓

1 → D[{3,4,5}][2] = 14 (2)
4 + 10 = 14 ✓

5. Consider the following puzzle problem: given a rectangular grid with row totals, column totals, and a cost in the upper-left corner of each cell, determine the lowest cost way to place items in the cells so that the total number of items placed in each row adds up to the row total, the total number of items placed in each column adds up to the column totals, and the cost is computed by adding up the costs of the items in the cells, where for each cell the cost is the number in the upper-left corner times the number of items (where empty cells correspond to 0 items in the cell).

Consider this proposed greedy algorithm for solving an instance of this puzzle:

```
Repeat:
    find the cell with the smallest cost number (written in the upper-left corner of each cell)
    that is empty, and for which there are non-zero amounts left in its row and column
        write in that cell the largest number of items you can, subject to not violating the constraints that
        the row and column totals are less than or equal to the given numbers
until the amounts written fulfill all the row and column totals
```

For example, on this instance:

	1	2	3	4	5	
1	3	7	11	4	2	10
2	5	9	4	2	8	15
3	6	1	9	4	7	12
	8	6	10	7	6	

row totals

col totals

we would find the cell with cost 1 and put as many items as possible in it, which is 6:

	1	2	3	4	5	
1	3	7	11	4	2	10
2	5	9	4	2	8	15
3	6	1 6	9	4	7	12
	8	6	10	7	6	

Then we would use the cell with cost 2 (to break ties, use the uppermost row, and then the leftmost column, among cells with the same costs), and put 6 items in it, obtaining:

	1	2	3	4	5	
1	3	7	11	4	2 6	10
2	5	9	4	2	8	15
3	6	1 6	9	4	7	12
	8	6	10	7	6	

5. (continued) Then we would put 7 items in the next cheapest cell, obtaining:

	1	2	3	4	5	
1	^{3}	^{7}	^{11}	^{4}	^{2} 6	10
2	^{5}	^{9}	^{4}	^{2} 7	^{8}	15
3	^{6}	^{1} 6	^{9}	^{4}	^{7}	12
	8	6	10	7	6	

Next we would fill the cell with cost 3, obtaining:

	1	2	3	4	5	
1	^{3} 4	^{7}	^{11}	^{4}	^{2} 6	10
2	^{5}	^{9}	^{4}	^{2} 7	^{8}	15
3	^{6}	^{1} 6	^{9}	^{4}	^{7}	12
	8	6	10	7	6	

Then the next cheapest cell that is empty and still has unused items in its row and column totals is row 2, column 3, so we put 8 items in it:

	1	2	3	4	5	
1	^{3} 4	^{7}	^{11}	^{4}	^{2} 6	10
2	^{5}	^{9}	^{4} 8	^{2} 7	^{8}	15
3	^{6} 4	^{1} 6	^{9} 2	^{4}	^{7}	12
	8	6	10	7	6	

a. [3 points] In the chart above (or written somewhere separately is that's more convenient), finish this example of the algorithm on the given instance.

b. [5 points] Now you have to decide whether this algorithm always produces the optimal cost, or whether it can fail. If you think that the algorithm is correct, just say so. If you think that it can fail, find the simplest (fewest number of cells) example you can for which this greedy algorithm fails—draw the chart, show how the greedy algorithm fills it in, and then show that there is a lower cost way of choosing the amounts for the cells.

1	^{3} 8	^{7}	^{11}	10
2	^{5}	^{9}	^{4} 10	15
3	^{6}	^{1} 6	^{9}	12
	8	6	16	

The algorithm will work for the given 3x5 matrix using the given costs but if some verticies are taken away, it will fail.

Name Matt Hart

6. Consider this problem: given a directed, unweighted graph with vertices v_1, v_2, …, v_n, and no cycles, report the number of different paths from v_1 to v_n.

Your job is to use the dynamic programming approach to design an algorithm for this problem, following the suggested ideas (no credit for inventing your own algorithm following different ideas!).

Use a one-dimensional chart, where $N[j]$ is the number of different paths from v_1 to v_j. With this idea, $N[1]$ is the number of different paths from v_1 to v_1, which is the base case for the recursion, and is taken as 1.

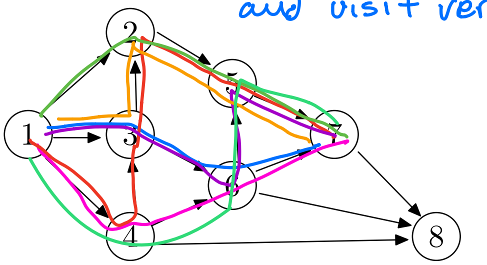
a. [1 point] What cell in the chart gives the answer to the entire problem?

4

b. [3 points] Figure out and explain the key recursive idea, namely how can cell $N[j]$ be computed assuming that the required cells $N[k]$ for other vertices v_k have already been computed? Note that your answer to this will need to refer to the graph.

The traversing will start from the source node and push s in queue. s will be marked "visited". The first iteration, s will pop from the queue. Neighbors N[1], N[2], N[3] will be visitied in the order in which they were created. The vertices that are visited pushed in the queue. when the queue is empty it will break when all nodes have been visited using a Bredth First Search. 1. Use top down approach and visit all nodes that are in layer 1 and move to the next and visit vertices in that layer.

c. [4 points] Demonstrate the full algorithm by filling in the empty dynamic programming chart given below for this graph:



1	2	3	4	5	6	7	8
1	3	2	1	7	4	6	5

As part of this calculation, to make sure that you are using the recursive idea and not just brute force counting, state *precisely* how $N[8]$ is computed in terms of other cells of the chart:

Name Matt Hurst

7. Using your computer and the ManualSimplex program, solve the instance of LP given below by the brute force method. Fill in the given chart with the basic variables and objective function value for all choices of basic variables that give a feasible point, and then state the optimal choice of basic variables with their values and the optimal objective function value.

$$\begin{aligned}\max\quad & -3x_1 + 2x_2 - 4x_3 - x_4 - 5x_5 + 7x_6 \quad \text{subject to}\\ & 2x_1 + x_2 - 3x_3 + 4x_4 - 5x_5 + 6x_6 = 15\\ & x_1 + 4x_2 + 6x_3 + x_4 + 2x_5 - 3x_6 = 3\\ & 6x_1 + 4x_2 + 2x_3 + 9x_4 + x_5 - 5x_6 = 10\\ & x \geq 0\end{aligned}$$

Basic Variables	Objective Function Value
x_1 x_2 x_6	7.8456376
x_1 x_3 x_6	1
x_2 x_4 x_5	-17.6
x_2 x_4 x_6	11.780488
x_3 x_4 x_6	6.3684211
x_4 x_5 x_6	-4.4

(may need more or less rows, of course)

Optimal basic variables and their values:

x_2, x_4, x_6

$x_2 = 1.5121951$

$x_4 = 1.2317073$

$x_6 = 1.4268293$

Optimal objective function value:

$z + 1.9(x_2) + 4.9756(x_3) + 0.8699(x_5) = 11.780488$