November 29, 2022 cmput304 seminar guide

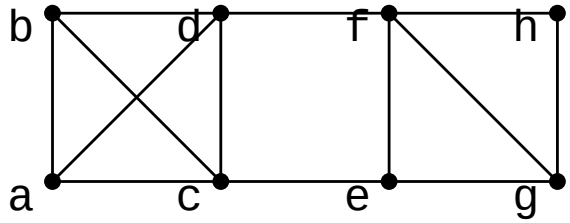
seminar 8

A *cut* of a graph is a partition of the node set into two non-empty subsets, called *parts*. A graph with only one node has no cuts (why?). A graph with node set $\{a, b\}$ has exactly one cut: $\{\{a\}, \{b\}\}$. A graph with node set $\{a, b, c\}$ has exactly three cuts: $\{\{a\}, \{b, c\}\}$, $\{\{a, b\}, \{c\}\}$, $\{\{a, c\}, \{b\}\}$. The *size of a cut* is the number of edges that cross the cut. A *min cut* is a cut with minimum size.

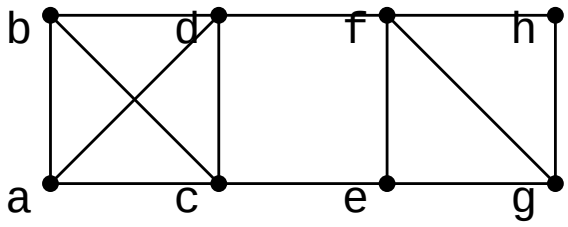
For a graph G with node set $V = \{v_1, \ldots, v_n\}$, define $Z(n)$ as the number of cuts. $Z(1)$ is 0, $Z(2)$ is 1, and $Z(n) = 2 * Z(n-1) + 1$ for $n \geq 3$. Why?

Well, put node v_n into one part and all other nodes into the other part: that is one cut. Now take any cut $Q = \{X, Y\}$ of $G - v_n$. You can add v_n to the part of Q containing v_1 or you can add v_n to the part of Q not containing v_1. This generates all possible cuts of G.

1. Prove by induction: $Z(n) = 2^{n-1} - 1$. Hint: use the above relation.
2. For the graph below, give a) the number of cuts b) all min cuts.



3. Let G be a cycle with $n \geq 3$ nodes and C a cut of G. a) Give all possible sizes of C. b) Give the size of a min cut of G.
4. Let C be the size of a min cut of a connected graph G. Prove/disprove: G has a min cut $\{X, Y\}$ in which both $G[X]$ (the subgraph of G obtained by removing all nodes not in X and removing all edges with an end not in X) and $G[Y]$ are connected.



5. For this graph, give a a) not-min cut that will never be found by the RKMC algorithm b) not-min cut that can be found by RKMC.

6. On this graph, trace 3 executions of RKMC with these edge permutations: a) alphabetic order, e.g. (ab, ac, ad, bc, bd ... gh) b) reverse alphabetic order, e.g. (hg, hf, gf, ge ... ba) c) (ac, ce, eg, gh, fh, df, bd, ...). For each execution, show the final forest, give the cut, and its size.

7. a) For this graph, for one execution of RKMC with uniform-random edge order, what bound does the theorem we saw in class guarantee on the probability that the RKMC cut is a min cut?

   b) In this case, is the bound from a) exact? Justify briefly.

8. For this graph, using the bound from the theorem, how many executions of RKMC would you need to run in order to guarantee that the probability of finding a min cut is at least .9?

seminar 7

1. Trace the DPV algorithm on this knapsack instance:

```
n 8   W 30
wts      9   8   7   6   5   4   3   4
vals     2   3   4   5   6   7   8   9
    none 1   2   3   4   5   6   7   8   <- item index
val  0   0   0   0   0   0   0   0   0
     1   -   -   -   -   -   -   -   -
     2   -   -   -   -   -   -   -   -
     3   -   -   -   -   -   -   -   -
     4   -   -   -   -   -   -   -   -
     5   -   -   -   -   -   -   -   -
     6   -   -   -   -   -   -   -   -
     7   -   -   -   -   -   -   -   -
     8   -   -   -   -   -   -   -   -
     9   -   -   -   -   -   -   -   -
    10   -   -   -   -   -   -   -   -
   ...               ...
```

   a) Show rows value-0 through value-10 of the final matrix. Recall: each row and columns correspond to values and item subsets respectively.

   b) Give the number of rows in the final matrix.

   c) Give an optimal solution.

2. In big Theta notation, give the worst-case runtime for the knapsack DPV algorithm. Assume that the input consists of n items and that each weight and each value is an n-bit integer, e.g. if $n = 10$ then each weight and each value is an integer in $\{2^9, 2^9 + 1, \ldots, 2^{10} - 1\}$. Express your answer as a function of n.

3. Suppose we use the knapsack DPV FPTAS to find an approximate solution with $\varepsilon = .1$ for this knapsack instance:

```
n 20   W 200
wt      21   14   27   10   21   25   28   14   25   14
        11   24   30   11   26   13   20   11   17   30
val   7486 5072 7810 4915 6186 5580 6202 7681 5061 5010
      7677 5436 5803 6643 5652 6350 5371 6650 5671 8000
```

   What is the scaling factor?

   Give the knapsack instance of the scaled problem: n? W? weights? values? (give the first 3)

   Assume that this is the solution to the scaled problem: blah Give the corresponding approximate solution to the original problem. Give the range of possible values for the optimal value of a solution to the original problem.

4. Assume that you are implementing the 0-1 knapsack FPTAS for a particular knapsack instance and ε.

   a) Assume that the scaling factor $n/(\varepsilon v_{\texttt{max}})$ is exactly 2 and that the you implement the algorithm as usual. In words, describe what happens.

   b) In general, if the scaling factor is greater than or equal to 1, how should you proceed? Explain briefly.

5. Page 281 of the Dasgupta et al. text gives a proof of correctness of the 0-1 knapsack FTPAS. The proof has 1 equation and 5 inequalities. Explain why each of these 6 relations holds.

seminar 6 (part b)

1. Let G be the weighted graph from the tsp-approx handout:

   `http://webdocs.cs.ualberta.ca/~hayward/204/jem/pix/tsp.pdf`

   Recall that we call an MST *Kruskal* if it is found by some execution of Kruskal's algorithm. Recall that every MST is Kruskal. Give the number of MSTs of G. Explain briefly.

2. In the tsp-approx handout, assume that the starting Eulerian tour is ACEGEFHIHFJFEDBDECA.

   a) Give the Hamiltonian cycle found by the 2-approx method.

   b) Give the Hamiltonian cycle found by the 1.5-approx method.

3. Let G be a weighted complete graph with at least 4 nodes and positive edge weights. Let v be a node in G and let H be the graph obtained by removing v and all edges incident with v. Let h be the length of a min-weight Hamiltonian cycle in H. Let g be the length of a min-weight Hamiltonian cycle in G. Prove/disprove: $h \leq g$.

4. Recall: we say that a weighted graph G satisfies the *triangle inequality* if for each triangle in G with edges a, b, c, $\text{wt}(a) \leq \text{wt}(b) + \text{wt}(c)$. E.g. the triangle with edge weights 1,2,3 satisifies the triangle inequality. E.g. the triangle with edge weights 1,1,3 does not satisfy the triangle inequality. Repeat the previous question, assuming that G satisfies the triangle inequality.

5. Let G be a weighted complete graph with at least 4 nodes that has positive edge weights and satisfies the triangle inequality. Let T be an MST of G. Let Z be the odd-degree nodes in g. Let H be the complete subgraph of G induced by Z. Let g be the min weight of a Hamiltonian cycle of G. Let h be the min weight of a Hamiltonian cycle of H. Prove/disprove: $h \leq g$.

6. For the 2-approx example, prove that each shortcutting step yields a Hamiltonian cycle with weight less than or equal to the original Hamiltonian tour. I have started the answer: you finish it.

```
starting Hamiltonian tour:
A C E D B D E G E F J F H I H F E C A

tour after 1st part of 1st shortcut:
A C E D B   E G E F J F H I H F E C A
by the triangle inequality, this tour's length is not longer
than the length of the preceding tour, because:
  from triangle BDE, edges BD DE have been replaced with edge BE

tour after 2nd part of 1st shortcut:
A C E D B     G E F J F H I H F E C A
by the triangle inequality, this tour's length is not longer
than the length of the preceding tour, because:
...

tour after 2nd shortcut:
...

tour after 3rd shortcut:
...

tour after 4th shortcut (this is the Hamiltonian cycle):
...
```

seminar 6

1. a) Trace the greedy set cover algorithm on the instance below. In case of a tie, always pick the set with smaller index. E.g. if there is a tie between picking S9 and S13, pick S9.

   b) From the text, give an upper bound on the size of the set cover that will be found in a).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
S 0	*	*	-	*	-	-	-	-	-	-	-	-	-	-	-	*	-	-	-	*	*
S 1	-	*	-	-	*	-	-	-	*	-	-	-	-	-	-	*	-	-	-	-	-
S 2	-	-	-	-	-	*	-	-	-	*	*	*	-	*	-	-	-	-	-	-	*
S 3	-	-	*	-	*	-	*	-	-	-	-	*	*	*	-	-	-	-	-	-	-
S 4	-	-	*	-	*	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-
S 5	-	-	-	*	-	-	*	-	-	-	-	-	-	-	-	-	*	-	-	*	
S 6	-	-	-	-	*	*	-	*	-	-	-	-	-	-	*	-	-	-	-	-	
S 7	-	-	-	*	-	-	-	-	-	-	-	-	-	*	*	-	*	-	-	*	-
S 8	-	*	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-	*	*	-	-
S 9	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-	-	-	-	-	*
S10	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-	*	-	-	*
S11	-	-	-	*	-	*	-	-	-	-	*	-	-	-	-	*	*	-	-	-	
S12	-	-	-	*	-	-	-	-	*	-	*	-	-	-	-	-	-	-	*	-	
S13	*	-	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	*	-	-	
S14	-	-	-	-	-	*	-	-	-	-	-	-	*	*	-	*	-	-	-	-	

2. Consider the greedy set cover algorithm and analysis from §5.4 of Dasgupta et al.. Assume that $n = 100$ and $k = 10$. In a), b), c) do not use the claim in the text: instead, use the reasoning from the proof of the claim.

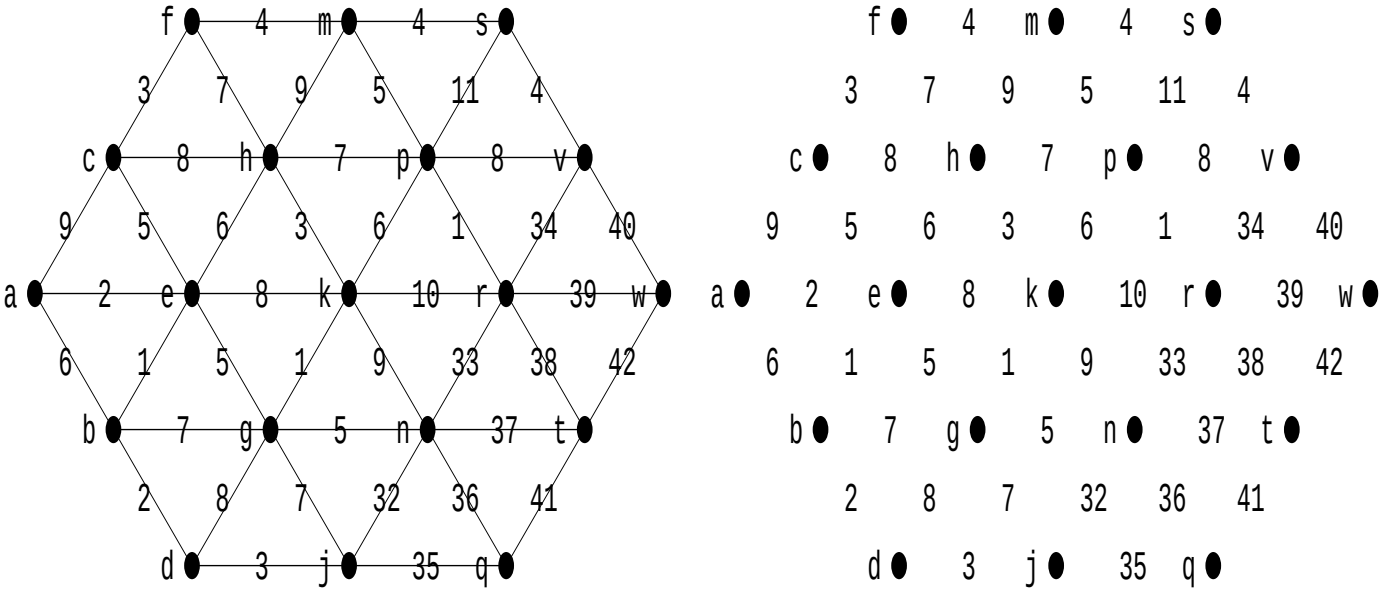
   a) Prove disprove: $n_1 \leq 90$.

   b) Prove disprove: $n_2 \leq 81$.

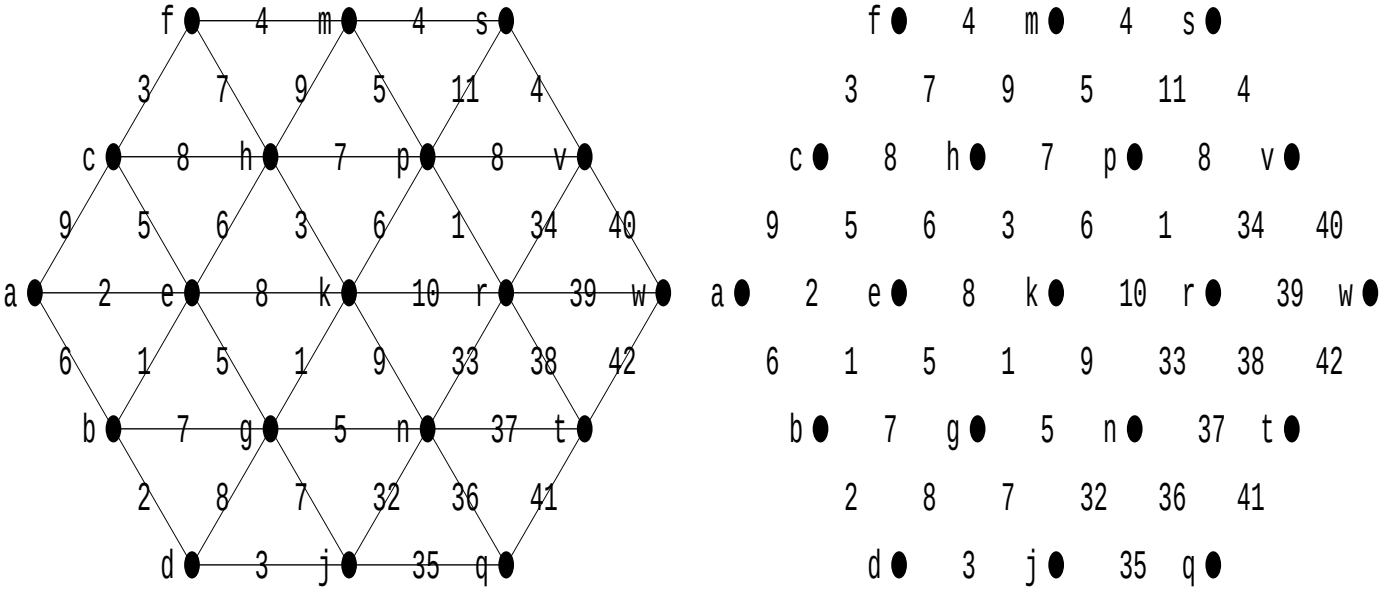
   c) Prove disprove: $n_3 \leq 72$.

   d) By the claim in the text, give an upper bound on the size of the set cover that will be found for this instance.

3. Draw the edges of a greedy sssp-tree from node f. In case of a tie when picking an edge, always pick the edge by alphabetic order, e.g. among { (mn), (dz) } pick (dz).

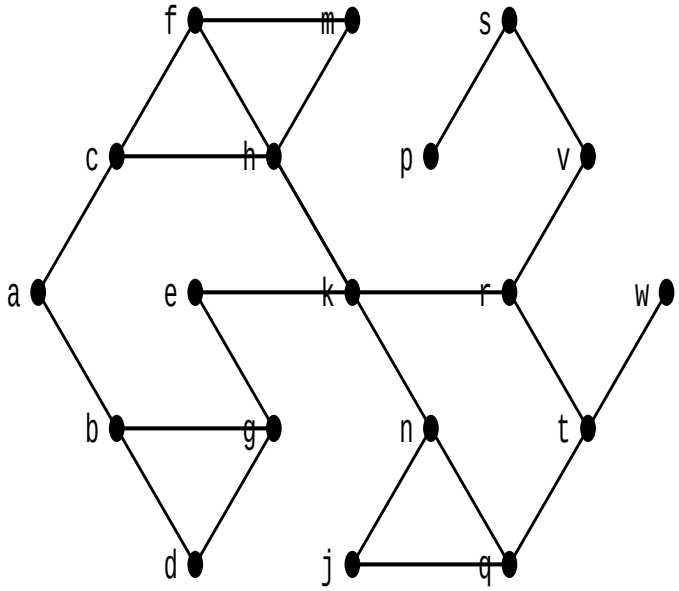


4. Draw the edges of a greedy (by-edge-weight, a.k.a. Kruskal) mst. When picking an edge, break ties alphabetically.



5. a) Trace the greedy set cover algorithm on the school location problem whose graph is below. Break ties alphabetically. At each step, list all the best options. Answer like this.

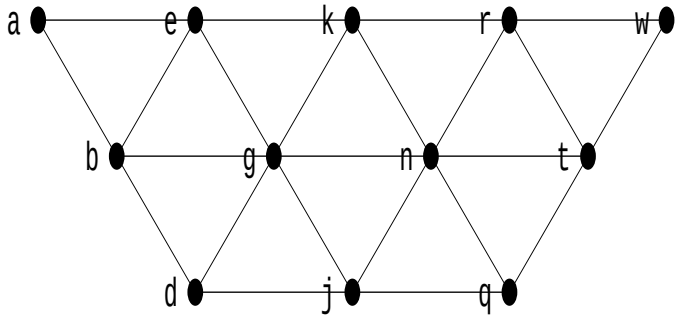
step	#uncovered towns	best town choice(s)	choice
1	19	???	?
2	?	???	?
3	?	???	?
...	...	...	...



b) Does the greedy algorithm find a min cover in this case? Prove your answer.

6. A *vertex cover* of a graph $G = (V, E)$ is a subset C of V that covers all edges. k-vertex cover is this problem: given a graph G and an integer k, does G have a vertex cover of size at most k?

   Find a minimum size vertex cover for this graph.



7. a) Prove that complement of a vertex cover (the set of nodes that are not in the cover) is an independent set.

   b) Prove that k-vertex cover is in the class NP-complete. Hint: use a) and the fact that k-independent set is in the class NP-complete.

   c) Explain why k-vertex cover is a special case of set cover.

   d) Prove that k-set cover is in the class NP-complete. Hint: use b) and c).

seminar 5

1. A. Give the dual of this linear program (LP):

$$\begin{array}{lrcrcrcr} \max & 4x_1 & + & 3x_2 & + & x_3 & & \\ \text{s.t.} & & & x_2 & + & 3x_3 & \leq & 5 \\ & 2x_1 & & & - & 4x_3 & \leq & -1 \\ & x_1 & + & x_2 & + & x_3 & \leq & 6 \end{array}$$

B. Give a short proof that the optimal value of the original LP is at most 16.

Here are two 2-player 0-sum matrix games. Each entry shows the payoff for the row player R.

```
 1  -1   2          0   4
 0   3   4          1   0
 2   1   0
```

2. In the game above left, R plays the mixed strategy (.5,.3,.2).

   a) When R plays this mixed strategy, what is R's guaranteed (i.e. minimax) payoff?

   b) Against this mixed strategy, what is column player C's best pure strategy?

3. Solve the game above right and prove that your answer is correct:

   a) Formulate the game as an LP,

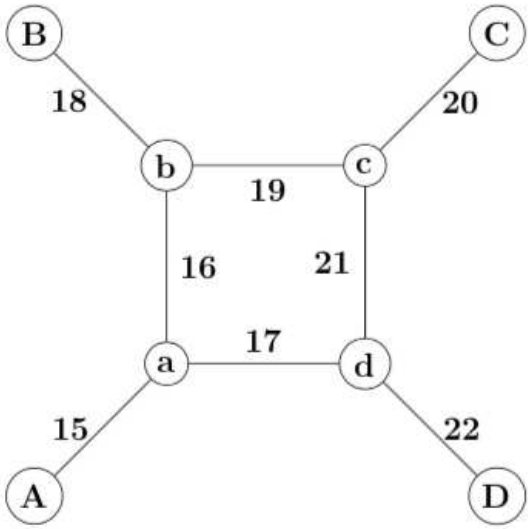
   b) give the equilibrium value,

   c) give a best-guarantee mixed strategy for R,

   d) give a best-guarantee mixed strategy for C,

   e) explain how you know that your answers to c) and d) are correct.

4. You manage a communications network with users A,B,C,D and bandwidths shown in the figure below. You need to establish a connection between each pair of users except for B and D (they never communicate). Connections A-B, A-C, A-D, B-C, C-D, pay 5, 2, 3, 1, 4 dollars respectively per unit bandwidth. Between each pair of users, at least 3 units must be routed.



There are two possible routes for every connection. For connection A-B, let xAB be the traffic volume routed A-a-b-B (the short way) and yAB the volume routed A-a-d-c-b-B (the long way). Define xBC, yBC, xCD, yCD, xAD, yAD similarly. Let xAC be the volume routed A-a-b-c-C and yAC the volume routed A-a-d-c-C.

You want to maximize this network's revenue. Using the variables defined above, formulate this problem as a linear program:

a) Give the objective function.

b) Give the system of (in)equalities.

c) Give a feasible solution.

You do not need to find an optimal solution.

seminar 4

1. For the LP below:

   a) rewrite it in vector format, as on page 209 of the Dasgupta text

   b) give the dual of the LP below, as on page 209

   c) sketch the feasible region and list all extreme points

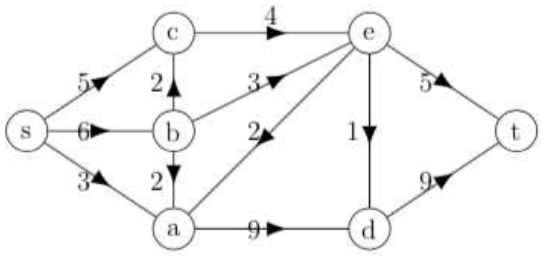
   d) give an optimal solution and value;

   answer like this: (0, 3, -.5) 39.5

$$
\begin{aligned}
\max\ 12x_1 + 6x_2 + x_3 & \\
x_1 &\leq 200 \\
x_2 &\leq 300 \\
x_1 + x_2 + x_3 &\leq 400 \\
x_2 + x_3 &\leq 400 \\
x_1, x_2, x_3 &\geq 0
\end{aligned}
$$

2. Using the residual network method, find a max flow and a min cut for this network. Show your work, as in the textbook.

   Give your final answer like this: `8 {s, a, b, d, t}`



seminar 3

1. *k-clique* is the following problem.

   input: a graph G and an integer k

   query: does G have a clique of size k?

   a) Define the problem *k-independent set*.

   b) Explain why the brute force method for solving k-clique runs in polytime if k is constant but in $\Omega(2^n)$ time when $k = n/2$.

   c) Give a polytime answer-preserving transformation T from k-clique to k-independent set (so T takes as input an instance of k-clique and gives as output an instance of k-independent set).

   Prove that T is polytime.

   Prove that T is answer-preserving.

2. a) Define the class of problems NP-complete.

b) So far in the lectures we have seen that these problems are NP-complete: sat (also called satisfiability); conjunctive normal form sat (cnf-sat); 3-cnf-sat, also called 3-sat; k-clique; k-independent set.

For each of these problems, explain briefly how we know that the problem is in the class NP-complete.

c) Does there exist a polytime answer-preserving transformation from sat to k-independent set? If yes, explain briefly how you know this. If no, explain briefly why not.

d) Is there a polytime answer-preserving transformation from k-independent set to sat? If yes, explain briefly how you know this. If no, explain briefly why not.

3. 2-coloring is this problem.

   input: a graph G.

   query: is there an assignment of at most 2 colors to nodes of G, such that each two nodes that are adjacent have different colors?

   a) Give a graph with 3 nodes that is not 2-colorable, and a graph with 6 nodes and six edges that is 2-colorable.

   b) Give a polytime answer-preserving transformation T from 2-coloring to sat.

   Prove that T is polytime.

   Prove that T is answer-preserving.

   c) Does b) imply that 2-coloring is in the class NP-complete? Explain briefly.

4. Let T be the transformation we saw in the lectures from 3-sat to k-independent set.

   a) For the 3-sat formula f below, draw the graph $T(f)$.

   $f = (x_1 \vee x_3 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \neg x_4)$.

   b) Let f be any satisfiable 3-sat formula with m clauses such that $T(f)$ has an independent set of size m. (i) Prove that $T(f)$ has no independent set of size $m + 1$ (ii) Prove that f is satisfiable.

seminar 2

1. The Fibonacci number $f(n)$ is defined as 0 if n is 0, 1 if n is 1, and $f(n-1)+f(n-2)$ for all integers $n \geq 2$. Prove by induction on j that, for all non-negative integers j, the value of a after line 4 has executed exactly j times is $f(j)$.

```
def ifib(n):              #line 0
  a,b = 0,1               #line 1
  for _ in range(n):      #line 2
    print(a)              #line 3
    a,b = b, a+b          #line 4
  return a                #line 5
```

2. Recall from class that the worstcase runtime of the usual addition algorithm is in $\Theta(n)$, where n is the size (number of bits) of the input.

   Show that the worstcase runtime of the usual multiplication algorithm is in $\Theta(n^2)$.

   Justify your answer carefully.

3. Recall from class: we say that the worstcase runtime complexity of addition is in $\Theta(n)$, because A) there is an algorithm that runs in $\Theta(n)$ time, and B) no algorithm runs in $o(n)$ time. In your own words, prove B.

4. Modify the LIS algorithm from class: for each index j, instead of storing only L[j], store LP[j], an ordered pair with first element L[j] and second element P[j], the predecessor of S[j] in some longest path ending at position j. Trace this LIS on the sequence below: show each update to LP.

   Here is an example.

j	0	1	2	3	4	5
S	3	0	5	2	4	6
LP	1,0					
		1,1				
			2,0			
				2,1		
					2,0	
					3,3	
						2,0
						3,2
						4,4

   Now trace the LIS on the sequence below.

j	0	1	2	3	4	5	6	7
S	7	2	9	4	10	6	8	5
LP								

5. Trace the dynamic-program-by-weight knapsack algorithm with this input:

```
val [7, 6, 10, 6, 9]
wt  [5, 8, 10, 8, 5]
W  23
```

6. Find a minimum set cover of these subsets:

```
    0 1 2 3 4 5 6 7 8 910111213141516171819
 S0  - - - - - * - - * - - - - - - - - - - -
 S1  - - - - - - - - - - - - - - - * - - * - -
 S2  - - - - * - - - * * - * - * * - - - - *
 S3  - - - - - - - - - - * * - * - - - - - - -
 S4  - - - - - - - * * - - - - * - - - - * -
 S5  - - * - - - - - - - - * - - - - - - - - -
 S6  - * * - - - * - * - - * - - - * * - * -
 S7  - - - - - - - - - - * - - - * * - - - * -
 S8  - - - * - - - - - - - * - - - - * - * - -
 S9  * - - - - - - - - - - - - * - - - - - - -
S10  - - - - - - - * - - - - - * - - * - - - - -
```

7. *Graph isomorphism* is this problem:

   Input. Two graphs $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$ with respective node sets V_0, V_1 and edge sets E_0, E_1.

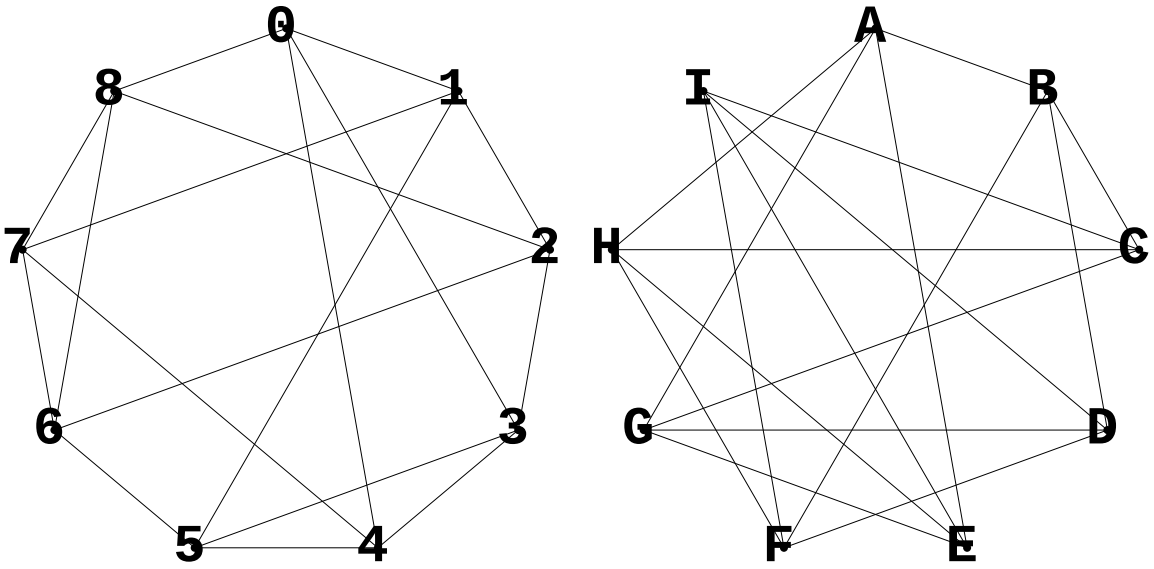
   Query. Is there an isomorphism $f : G_0 \to G_1$, i.e. a bijection $f : V_0 \to V_1$ such that $\{x, y\}$ is in E_0 if and only if $\{f(x), f(y)\}$ is in E_1.

   Explain why graph isomorphism is in the class NP.

8. Give a short proof that this function f is not an isomorphism between the two graphs below.

j	0	1	2	3	4	5	6	7	8
f(j)	A	B	C	D	E	F	G	H	I

9. Prove or disprove: the two graphs below are isomorphic.



seminar 1

1. Interpret the 4th and 5th digits of your student number as a 2-digit decimal number n. E.g. if your student number is ***70**, then n is 70. If n is square — 0, 1, 4, 9, 16, 25, 36, 49, 64, 81 — then increment n by 5. Now multiply n by 10^6. E.g. if your student number is ***70**, then n is now 70 00 00 00; if your student number is ***36**, then n is now 41 00 00 00.

   By hand, using the algorithm from class, find the largest integer t such that $t \leq \sqrt{n}$. Show your work.

2. For positive real numbers b and n with $b \neq 1$, $\log_b(n)$ is defined as the exponent x such that $b^x = n$. For positive numbers c, d, here is a proof of the log-of-product-rule, i.e. that $\log_b c + \log_b d = \log_b (cd)$.

   Proof. Let x, y, z be $\log_b c, \log_b d, \log_b (cd)$ respectively.

   $b^{x+y} = (b^x)(b^y)$ by the exponentiation-of-a-sum rule,

   $b^x = c$ by the definition of x,

   $b^y = d$ by the definition of y, so

   $b^{x+y} = (b^x)(b^y) = cd$, so

   $x + y = \log_b(cd)$ by the definition of $\log_b(cd)$. Also,

   $x + y = \log_b(c) + \log_b(d)$ by the defintions of x and y, so

   $\log_b c + \log_b d = \log_b cd$.

   This is the log-of-ratio rule: for positive numbers c, d with $\log_b c > \log_b d$, $\log_b (c/d) = \log_b(c) - \log_b(d)$.

   Prove the log-of-ratio rule.

3. Define $f(n)$ as 0 if n is 0, 1 if n is 1, and $f(n-1)+f(n-2)$ if n is an integer greater than or equal to 2. Consider this python procedure:

```
def fib(n):
  if (n<=1):
    return n
  return fib(n-1) + fib(n-2)
```

Claim. For all integers $n \geq 0$, `fib(n)` returns $f(n)$.

Proof. Argue by induction on n. The case when n is 0 or 1 was proved in the lecture.

Fix n to be some integer t greater than or equal to 2. Assume that the claim holds for all values of n in the set $\{0, 1, \ldots, t-1\}$. In order to complete the proof, we now want to show that the claim holds when n is t, i.e. that `fib(t)` returns $f(t)$.

So what happens when `fib(t)` executes? Well, $t \geq 2$, so the if condition evaluates to false, so the program returns `fib(t-1)+fib(t-2)`.

(now you complete the proof ...)

4. Let $f(n) = 1.6^n$. Let $g(n) = 2^n$. Complete the proof below. Do not use limits in your answer.

   Claim: $g(n)$ is not in $O(\ f(n)\ )$.

   Proof of claim, by contradition: Assume that $g(n)$ is in $O(f(n))$. Then by the definition of $O(f(n))$, there exists an integer n_0 and a positive constant c such that $g(n) < cf(n)$ for all integers $n \geq n_0$. So ...

5. Let $f(n) = 3n^2 + 5n \lg n$.

   a) Give the simplest function $g(n)$ such that $f(n) \in \Theta(g(n))$. Justify briefly (just a few words, you do not need a proof).

   b) As n gets large, the ratio $f(2n)/f(n)$ approaches a constant c: what is the constant? Does it approach this constant from below, from above, or does it oscillate? Justify briefly.