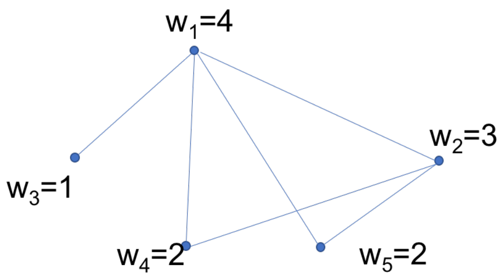
Exam Combinatorial Optimization, 23 October 2018.

- It is not allowed to use any books, notes, or calculator. Just pen and paper.
- The table shows the maximum number of points per (sub-)question.
- You may also obtain points for partial solutions.

1a	1b	1c	1d	2	3	4a	4b	5a	5b	6a	6b	6c	6d	7a	7b	Σ
1	3	3	2	6	4	3	3	2	4	2	2	2	2	2	4	45

---

1. The figure shows an instance of the *weighted* Vertex Cover problem (VC).



(a) What is the optimal value?

(b) Write down the ILP for this VC instance. Give a solution to the LP-relaxation of value 6.

(c) Write down the dual of the LP. (Use variable a y_{ij} for each edge (i,j).) Give a solution to the dual of value 6.

(d) Now, let $G=(V,E)$ be an arbitrary weighted graph where for each vertex v_i its weight w_i is equal to its degree d_i. (As is the case for the given graph.) Argue that the optimal solution to the LP-relaxation is exactly $|E|$ (the number of edges). *(Hint: Use here that if there is a solution for the LP and a solution for the dual of the same value, then both are optimal.)*

2. A *triangle* in a graph $G=(V,E)$ is a clique of size 3, that means, a set of vertices $v,w,z \in V$ such that the edges $(v,w),(w,z)$, and (v,z) exist in the graph.

TRIANGLE HITTING SET:

Instance:	A Graph $G=(V,E)$.
Solution:	A set $S \subseteq V$ such that each triangle has at least TWO vertices in S.
Value:	The number of vertices in S.
Goal:	Find a solution of minimum cost.

Give a 2-approximation algorithm for the Triangle Hitting Set problem using LP-rounding. (Give the algorithm and give a proof for the ratio of 2.)

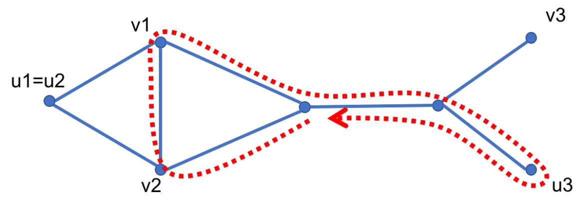
Hint: Define variables $x_i \in \{0,1\}$ for all $i \in V$. Denote by T the set of all triangles in the graph.

3. It is known that the Longest Processing Time algorithm is a $4/3$-approximation algorithm for the problem $P||C_{\max}$ (the standard scheduling problem on parallel machines.) Now assume that you apply it to some instance and you notice that in your schedule each machine contains at least 5 jobs. Argue that the length of the schedule that you found it at most $5/4$ times the optimal length. (*Hint: Consider the job k that completes last.*)

4. Consider the following variant of the TSP:

AT-LEAST-ONE-OF-TWO TSP (1OF2 TSP):

Instance:	Unweighted graph $G = (V,E)$ on n vertices, and pairs of vertices $(u_1,v_1),(u_2,v_2),\ldots,(u_k,v_k)$. (A pair does not need to be an edge of the graph.)
Solution:	Tour that visits *at least one* vertex from every pair (u_i,v_i).
Value:	Length of the tour.
Goal:	Find a tour of minimum length.



An example with $k = 3$ pairs. The optimal tour (dotted line) has length 7.

Clearly, 1OF2 TSP is NP-hard since it generalizes the standard TPS problem on unweighted graphs. (Take $u_i = v_i$ for all vertices $v_i \in V$.) In (a) and (b) you are asked to show that even for some special graphs, the problem remains NP hard. (The two questions can be answered independently.)

(a) Prove, by a reduction from Vertex Cover that 1OF2 TSP is NP-hard even for *complete graphs*. (In other words, assume that you have some polynomial time algorithm for 1OF2 TSP on complete graphs. Show how such an algorithm can be used to solve the VC in polynomial time on general graphs.)

(b) Prove, by a reduction from Vertex Cover, that 1OF2 TSP is NP-hard even for *trees-metrics*. (In other words, assume that you have some polynomial time algorithm that solves 1OF2 TSP on trees. Show how such an algorithm can be used to solve the VC in polynomial time on general graphs.)
(*Hint: Consider a star on $n+1$ vertices: One vertex has degree n and the other vertices have degree 1.)*

5. Section 4.1 of the book considers the problem of minimizing the total completion time on a single machine under release times constraints: $1|r_j|\sum C_j$.

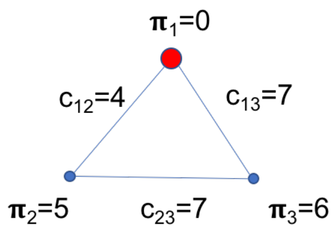
(a) Describe a 2-approximation algorithm for this problem. (No proof. Just describe.)

Section 4.2 of the book considers the weighted version of this problem: $1|r_j|\sum w_jC_j$ and presents a 3-approximation algorithm using LP-rounding.

(b) Describe this 3-approximation algorithm. That means, give the LP and explain how it is used to construct a schedule. (No proof needed, just describe the algorithm.)

6. **(a)** Give the ILP for the Prize Collecting Steiner Tree problem. (The general formulation.)

(b) Give the ILP for the Prize Collecting Steiner Tree problem for the given instance. (Vertex 1 is the root.) Also give (draw) the optimal solution for this instance.



(c) Explain what a separation oracle is.

(d) Argue that the ILP for Prize Collecting Steiner Tree does have a separation oracle that runs in polynomial time.

7. **(a)** Let G be a 3-colorable graph. Assume that you are given a 3-coloring of the graph. Show how to use it to find a cut that contains at least two thirds $(2/3)$ of the edges. *Hint: Consider 3 possible cuts and argue that one of those contains at least two thirds of the edges.*

(b) Now assume that you do not have the 3-coloring. You only know that it can be colored with 3 colors. Show that it is still possible to find a random cut such that the *expected* number of edges in the cut is at least $2/3$ of all the edges in the graph.
(*Hint: Give the Vector Program. Show that the optimal value of the VP is at most* -0.5*. Then describe your randomized algorithm. Argue that each edge appears in the cut with probability at least* $2/3$.)

Appendix

The following problems can be solved in polynomial time:

- Linear programs and Vector programs.
- Finding a minimum cut in a graph.
- Finding a maximum flow in a graph.
- Finding the shortest path between two points in a graph.
- Finding a minimum spanning tree in a graph.
- Finding a maximum matching in a graph.
- Finding a minimum cost perfect matching in a graph.