

General instructions. Answer the problems given in the rest of this document. Edit this LaTeX template to add your answers, as described in the file named `00_README.txt`, then compile it to produce `homework_10.pdf`.

Don't know? Say so. If you don't know the answer to a problem or any part of the problem, we encourage you to simply write “*I don't know*” instead of giving an answer. (If you write “I don't know” you can also add any questions you might have.)

Gentle reminder: homework is for learning. As noted in [Lecture Note 1](#), engaging fully with the homeworks is the best way to learn the material. *The primary role of homeworks is to help you learn the material, and the primary role of feedback on homeworks is to help you improve your ideas.*

Please try to explain your ideas as clearly as you can. The clearer your explanations are, the more useful the feedback on them can be.

Collaboration and citation policy. If you are doing this homework as part of a course, follow the course's collaboration and citation policy. In any case, at the end of each answer, list the people you collaborated with, and cite every source that your answer uses ideas from (other than the lecture notes and the sources listed in the lecture notes). If there are none, write “none”.

Problem 1. (Section 8.2) In the ruins of Pompeii, the [House of the Tragic Poet](#) has a famous mosaic floor proclaiming “Beware of the Dog.” In Boston, a less tragic and wealthier poet has commissioned a mosaic to be built from 1kg bars of solid gold, each 80mm by 40mm. To know whether her desired mosaic can be built in this way, the poet needs to solve the following problem:

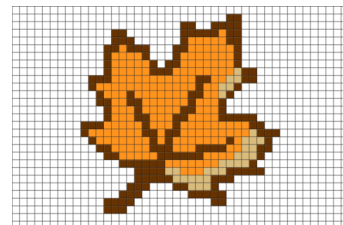
Tiling by Bars

input: a two-dimensional array $A[1..n, 1..m]$ where each entry $A[i, j]$ is either ‘colored’ or ‘white’, representing an $m \times n$ grid of square cells, each 40mm by 40mm and colored or white.

output: ‘yes’ if it is possible to place gold bars (each 80mm by 40mm) on the grid so that each bar covers exactly two colored grid cells, and each colored grid cell is covered by exactly one bar. If it is not possible, then ‘no’.



The gold bars can be rotated 90 degrees, but not split. As an example, consider the pixel art on the right (not to scale). Can the leaf be tiled in this way?



Develop a poly-time reduction r from this problem to the Bipartite Matching decision problem:

Bipartite Matching

input: bipartite graph $G = (U, W, E)$ and non-negative integer k

output: ‘yes’ if G has a matching of size at least k , otherwise ‘no’.

- Define your reduction r by defining, for every possible instance $A[1..n, 1..m]$ of Tiling by Bars, what output $r(A)$ the reduction gives. The output will be a bipartite graph $G = (U, W, E)$ and integer k . Your definition must state precisely how G and k are defined for the given A .¹
- Show that your reduction is a *polynomial-time* reduction, by explaining carefully how, given A , one can compute the output $r(A)$ in time polynomial in the input size, nm .
- For your reduction r to be correct, it must be that, for every Tiling by Bars instance A , the answer to A is ‘yes’ if and only if the answer to the Bipartite Matching decision problem $r(A)$ is ‘yes’. Show the “if” direction. Given an arbitrary Tiling by Bars instance A , let $G = (U, W, E)$ and k be the output of your reduction r given A . That is, $(G, k) = r(A)$. Describe carefully how, given any bipartite matching M in G of size at least k , one could obtain from M a valid covering of A by bars.
- Now show the “only if” direction. Given an arbitrary Tiling by Bars instance A , let $G = (U, W, E)$ and k be the output of your reduction r given A . Describe carefully how, given any valid covering of A by bars, one could obtain a bipartite matching M in G of size at least k .
- Let R be the algorithm from Part (b) above that computes your reduction r in polynomial time. Assume that there is a poly-time algorithm B that solves the Bipartite Matching decision problem. Give pseudo-code for an algorithm that uses R and B to solve the Tiling by Bars decision problem in polynomial time. Your algorithm may call B and R as subroutines. You don’t need to define B or define R . Your pseudo-code should have just a few short lines. *Hint: Section 8.2.4, Lemma 8.6.*

You can use notation such as $B(G, k)$ to denote executing B on a given instance (G, k) of Bipartite Matching, and returning the result (‘yes’ or ‘no’) as the value of the expression $B(G, k)$. You can use notation such as $(G, k) = R(A)$ to denote executing R on input A , and storing the result ($r(A)$) in G and k . (This problem is due to Jon Ullman.)

¹**Hint:** Imagine a checkerboard on the grid. Any bar must cover exactly one red square and one black square.

Problem 1 answer

(a) Given an instance $A[1..n, 1..m]$ of Tiling by Bars, the reduction r outputs the pair $(G, k) = r(A)$ where $G = (U, W, E)$ and k are defined as follows:

★...★

(b) Given an instance $A[1..n, 1..m]$ of Tiling by Bars, the output (G, k) of the reduction defined in Part (a) can be computed in time polynomial in the size of A , that is, nm , as follows:

★...★

(c) Fix an arbitrary TILING BY BARS instance $A[1..n, 1..m]$. Let $G = (U, W, E)$ and k be the output $r(A)$ of the reduction r given A . Consider any matching M in G of size at least k . From M , we can obtain a valid tiling T (of A by bars) as follows.

★...★

Problem 1 answer, continued

(d) Fix an arbitrary TILING BY BARS instance $A[1..n, 1..m]$. Let $G = (U, W, E)$ and k be the output $r(A)$ of the reduction r given A . Consider any valid tiling T of A by bars. From the tiling T , we can obtain a matching M of size at least k as follows.

★...★

(e) Here is the algorithm that uses R and B to solve TILING BY BARS:

TileWithBars($A[1..n, 1..m]$):

1. ★...★

2. ∴

Problem 1 collaboration and citations

I collaborated on this problem with the following people:

● ★...★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

● ★...★

Problem 2. (Section 8.3)

Find the optimal solution for each of the following linear programs:

- (a) maximize $x_1 + 2x_2$ subject to

$$2x_1 + x_2 \leq 10$$

$$x_1 + 3x_2 \leq 9$$

$$x_1, x_2 \geq 0$$

Hint: Since there are only two variables in the linear program, you can plot the constraints in two dimensions on paper, following the example in Section 8.3, Figure 8.5.

- (b) minimize $x_1 + 3x_2 + 5x_3$ subject to

$$x_1 + x_2 + x_3 \geq 1$$

$$x_1, x_2, x_3 \geq 0$$

Hint: Consider raising just one x_i to 1. Which one is cheapest (contributes least to the objective function)?

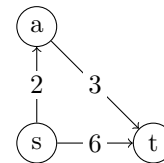
- (c) maximize $\pi_t - \pi_s$ subject to

$$\pi_a - \pi_s \leq 2$$

$$\pi_t - \pi_a \leq 3$$

$$\pi_t - \pi_s \leq 6$$

Hint: Apply Section 8.3.3, Lemma 8.12 to the graph below:

**Problem 2 answer**

- (a) One optimal solution is $(x_1, x_2) = (\star \cdots \star, \star \cdots \star)$

- (b) One optimal solution is $(x_1, x_2, x_3) = (\star \cdots \star, \star \cdots \star, \star \cdots \star)$

- (c) One optimal solution is $(\pi_s, \pi_a, \pi_t) = (\star \cdots \star, \star \cdots \star, \star \cdots \star)$

Problem 2 collaboration and citations

I collaborated on this problem with the following people:

• $\star \cdots \star$

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• $\star \cdots \star$

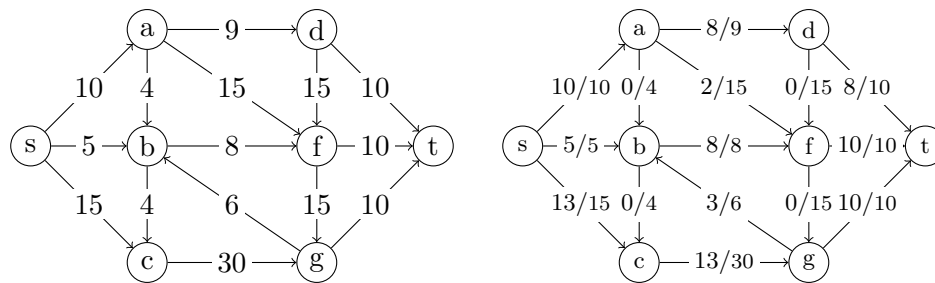
Problem 3 part (i). (Section 8.1) *Do this OR Problem 3 part (ii), not both.*

Define an edge (u', w') of an $s - t$ flow network $G = (V, E)$ to be *critical* if, for every $\epsilon > 0$ (no matter how small) decreasing the capacity of (u', w') by $\min(\epsilon, \text{cap}(u', w'))$ would decrease the maximum possible flow. That is, for all $\epsilon > 0$, in the flow network $G_\epsilon = (V, E)$ with edge capacities

$$\text{cap}_\epsilon(u, w) = \begin{cases} \text{cap}(u, w) & \text{if } (u, w) \neq (u', w') \\ \max(\text{cap}(u, w) - \epsilon, 0) & \text{if } (u, w) = (u', w'), \end{cases}$$

the maximum s - t flow value is strictly less than the maximum s - t flow value in G .

(a) List all critical edges in the flow network below (on the left, from Homework 9 Problem 4). For convenience, to the right is one possible max flow.



(b) Give a long-form proof of the following lemma.

Lemma 1. *Let (u, w) be any edge in any s - t flow network G with positive capacities. Then (u, w) is critical if and only if (u, w) crosses some minimum-capacity s - t cut (S, T) (that is, $u \in S$ and $w \in T$).*

You may use any lemmas or theorems from the lecture notes, such as the Max-Flow/Min-Cut Theorem. Please cite them clearly when you do.

Problem 3 part (i)(a) answer Here is a comma-separated list of all the critical edges:

★...★

Problem 3 part (i)(b) answer

Lemma 1. *Let (u, w) be any edge in any s - t flow network G with positive capacities. Then (u, w) is critical if and only if (u, w) crosses some minimum-capacity s - t cut (S, T) (that is, $u \in S$ and $w \in T$).*

Proof (long form).

1. Consider any edge (u, w) in any s - t flow network G with positive capacities.
2. Let $c_{\min}$ be the minimum capacity of any s - t cut in G .
3. By Theorem 8.2 (Max-Flow Min-Cut), the maximum s - t flow value in G is $c_{\min}$.
4. Let c_{uw} be the minimum capacity of any s - t cut in G that (u, w) crosses. Clearly $c_{\min} \leq c_{uw}$.

5. First we prove that, if (u, w) doesn't cross any min-capacity cut, then (u, w) is not critical.
 - 6.1. Suppose that (u, w) doesn't cross any min-capacity s - t cut, that is, $c_{\min} < c_{uw}$. *(block: if-then)*
 - 6.2. Fix $\varepsilon = \min(\text{cap}(u, w), c_{uw} - c_{\min})$. Note $\varepsilon > 0$.
 - 6.3. Let G_ε be the flow network obtained by reducing the capacity of (u, w) by $\min(\varepsilon, \text{cap}(u, w))$.
 - 6.4.1. Consider an arbitrary s - t cut (S, T) . *(block: $\forall$)*
 - 6.4.2. Let $\text{cap}(S, T)$ and $\text{cap}_\varepsilon(S, T)$ denote the capacities of (S, T) in G and G_ε , respectively.
 - 6.4.3. $\star \cdots \star$
 - 6.4.4. $\vdots$
 - 6.4.5. So $\text{cap}_\varepsilon(S, T)$ is at least $c_{\min}$.
 - 6.5. By Block 6.4, the capacity of every s - t cut in G_ε is at least $c_{\min}$.
 - 6.6. So, by Thm. 8.2 (Max-Flow Min-Cut), the max flow value in G_ε is at least $c_{\min}$.
 - 6.7. So the max flow value in G_ε is at least the max flow value in G .
 - 6.8. So (u, w) is not critical.
7. By Block 6, if (u, w) doesn't cross any minimum-capacity cut, then (u, w) is not critical.

8. Next we prove that, if (u, w) crosses some minimum-capacity s - t cut, then (u, w) is critical.
 - 9.1. Suppose that (u, w) crosses some such cut (S, T) (with $\text{cap}(S, T) = c_{\min}$). *(block: if-then)*
 - 9.2.1. Consider an arbitrary $\varepsilon > 0$. *(block: $\forall$)*
 - 9.2.2. Let G_ε be obtained from G by reducing the capacity of (u, w) by $\min(\varepsilon, \text{cap}(u, w)) > 0$.
 - 9.2.3. Let $\text{cap}(S, T)$ and $\text{cap}_\varepsilon(S, T)$ denote the capacities of (S, T) in G and G_ε , respectively.
 - 9.2.4. $\star \cdots \star$
 - 9.2.5. $\vdots$
 - 9.2.6. So the max s - t flow value in G_ε is strictly less than the max s - t flow value in G .
 - 9.3. By Block 9.2, for all $\varepsilon > 0$, reducing (u, w) 's capacity by $\min(\varepsilon, \text{cap}(u, w))$ reduces the max flow.
 - 9.4. So (u, w) is critical.
10. By Block 9, if (u, w) crosses some minimum-capacity cut, then (u, w) is critical.

11. By Steps 7 and 10, (u, w) is critical if and only if (u, w) crosses some minimum-capacity cut. $\square$

Problem 3 part (ii). (Section 8.1) *Do this OR Problem 3 part (i), not both.*

Give a long-form proof of the conjecture below, suggested by a student in Tuesday's class:

Conjecture 1. *Let f be any maximum flow in any $s - t$ flow network G . Let R be the set of vertices reachable from s in the residual graph G_f . Let (S, T) be **any** minimum-capacity s - t cut. Then $R \subseteq S$. That is, every vertex reachable from s in G_f is in S .*

Recall that $(R, V \setminus R)$ is itself a min-capacity s - t cut. (See e.g. the proof of Theorem 8.2 in Section 8.1.4.) One interesting consequence of the conjecture is that, for any two max flows f and f' , the set of vertices reachable from s in G_f is the same as the set of vertices reachable from s in $G_{f'}$, even if f and f' are different flows.

In your proof, use the following notation, as defined in Section 8.1:

$|f|$: The value of a given flow f .

G_f : The residual graph for a given flow f .

$E(A, B)$: The edges in E from A to B ($E(A, B) = \{(u, w) \in E : u \in A, w \in B\}$).

$f(A, B)$: The flow from A to B (that is, the total flow on edges in $E(A, B)$).

$\text{cap}(S, T)$: The capacity of an s - t cut (S, T) (that is, the total capacity on edges in $E(S, T)$).

And use the following observation, from the proof of Lemma 8.2:

Observation 1. *For any s - t flow f and s - t cut (S, T) , $f(S, T) \leq \text{cap}(S, T)$ and $f(T, S) \geq 0$.*

Problem 3 part (ii) answer

Proof of Conjecture 1.

1. Consider any f , G , R , and (S, T) as described in the conjecture.
2. By Theorem 8.2, ★...★
3. By Lemma 8.1, ★...★
4. By Observation 1, ★...★
5. ★...★
6. ⋮
7. That is, every vertex that is reachable from s in G_f is in S . □

Problem 3 collaboration and citations

I collaborated on this problem with the following people:

• ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• ★ ⋯ ★