Problem 1. (10 points) Label each of the following statements True or False.

Every directed graph with $n$ vertices and $n(n-1)/2$ edges has a directed cycle.  \text{FALSE}

For any network the maximum cut is equal to the maximum flow.  \text{FALSE}

The set of languages accepted by 2-tape Turing Machines is the same as the set of languages accepted by 1-tape Turing Machines.  \text{FALSE}

Every Turing Machine accepts a recursively enumerable language.  \text{TRUE}

HALTING is a recursive language.  \text{FALSE}

Every recursive language is recursively enumerable.  \text{TRUE}

$\exists$ a recursively enumerable language whose complement is also recursively enumerable.  \text{TRUE}

If there is a reduction from PROB to HALTING then PROB must be undecidable.  \text{FALSE}

BIPARTITE MATCHING is NP-complete.  \text{FALSE}

There is a polynomial time reduction from 3SAT to VERTEX COVER.  \text{TRUE}


Problem 2. (10 points) Assume the graph on the left shows a network (with capacities) and the graph on the right shows the current flow in the network, construct the derived network and find an augmenting path.

Answer 2. The graph on the left is the derived network. The path on the right is the augmenting path.
Problem 3. (10 points) Show that the problem below is not decidable.

\[ L = \{ (M; w) : M(w) \text{ halts after an even number of steps} \} \]

Answer 3. Given an input to HALTING \((M; w)\) create an input to \(L (M'; w')\) where \(w' = w\) and \(M'\) (on any input) simulates \((M; w)\) and keeps track of the parity of the number of steps and (if it terminates) then either immediately halts or performs one final step and halts. \(M'\) chooses whichever of these causes the number of steps to be even.

Problem 4. (10 points) For each boolean expression either give a satisfying truth assignment or state that it is unsatisfiable.

\(\text{a) } (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_4) \land (\overline{x_2} \lor x_1 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_4) \)
\(\text{b) } ((\overline{x_1} \land x_2) \lor x_3) \land ((\overline{x_1} \lor x_2) \land x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor x_2 \lor \overline{x_3}) \land ((\overline{x_1} \land x_2 \land x_3) \lor (\overline{x_2} \land x_1 \land x_4)) \)
\(\text{c) } (x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_2} \lor x_1 \lor x_4) \)

Answer 4.

\(\text{a) } x_1 = T, x_2 = F, x_3 = F, x_4 = T \text{ is a satisfying truth assignment.} \)
\(\text{b) No satisfying truth assignment.} \)
\(\begin{array}{cccc|c}
  x_1 & x_2 & x_3 & x_4 & \text{expression} \\
  \hline
  T & T & T & T & F \\
  T & T & T & F & F \\
  T & T & F & T & F \\
  T & T & F & F & F \\
  T & F & T & T & F \\
  T & F & T & F & F \\
  T & F & F & T & F \\
  T & F & F & F & F \\
  F & T & T & T & F \\
  F & T & T & F & F \\
  F & T & F & T & F \\
  F & T & F & F & F \\
  F & F & T & T & F \\
  F & F & T & F & F \\
  F & F & F & T & F \\
  F & F & F & F & F \\
  \end{array} \)
\(\text{c) } x_1 = T, x_2 = F, x_3 = F, x_4 = T \text{ is a satisfying truth assignment.} \)

Problem 5. (10 points) Recall that SUBSETSUM is NP-complete. Show that the language EQUAL is also NP-complete.

\[ \text{EQUAL} = \{ S : S \text{ is a set of integers which can be split into two sets which sum to the same value} \} \]
Answer 5. Given an instance \((S, t)\) of \textit{SubsetSum} where \(S\) is a set of numbers and \(t\) is the target create an instance of \textit{Equal} by setting \(S' = S + \{s\}\) where \(s = |2t - \sum S_i|\).

If \(S\) has a subset that sums to exactly \(t\) then \(S\) can be split up into a set that adds up to \(t\) and a set that adds up to \(\sum S_i - t\) by adding \(s\) to the smaller of these both sets have the same sum (this sum is either \(t\) or \(\sum S_i - t\) depending on whether \(t\) is more or less than half the total).

If the \(S'\) created has two subsets that sum to exactly the same value then they either sum to \(t\) or \(\sum S_i - t\). In the first case, the set that does not include \(s\) is a subset of \(S\) that sums to \(t\). In the second case, removing \(s\) from the set that contains \(s\) results in a subset of \(S\) that sums to \(t\).

Problem 6. (10 points) The \textit{BinPacking} problem asks whether there is a way to split a set \(S\) of integers into \(b\) sets each of which sums to at most \(t\). Show that \textit{BinPacking} is NP-complete.

\textit{BinPacking} = \{(S; b, t): S can be split into \(b\) sets each of which sums to at most \(t\)\}

Answer 6. The previous problem showed that \textit{Equal} was NP-complete. Given an \(S\) an instance of \textit{Equal} create an instance of \textit{BinPacking} \((S; 2; t)\) where \(t = \frac{\sum S_i}{2}\).

If \(S\) has two equal sized subsets then they can be put into two bins with capacity half the sum. If \(S\) doesn’t have two equal sized subsets then they cannot be put into two bins with capacity half the sum.