CECS 528, Learning Outcome Assessment 8, 11/13/2024, Dr. Ebert

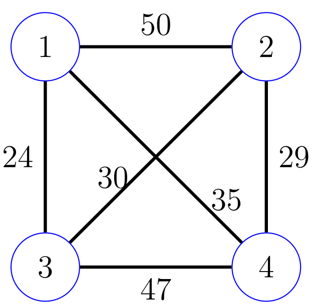
Problem

LO4. Do/Solve the following.

(a) The dynamic-programming algorithm that solves the `Runaway Traveling Salesperson` optimization problem (Exercise 30 from the Dynamic Programming Lecture) defines a recurrence for the function $\mathrm{mc}(i, A)$. In words, what does $\mathrm{mc}(i, A)$ equal? Hint: do *not* write the recurrence (see Part b). Hint: we call it "Runaway TSP" because the salesperson does *not* return home.

(b) Provide the dynamic-programming recurrence for $\mathrm{mc}(i, A)$.

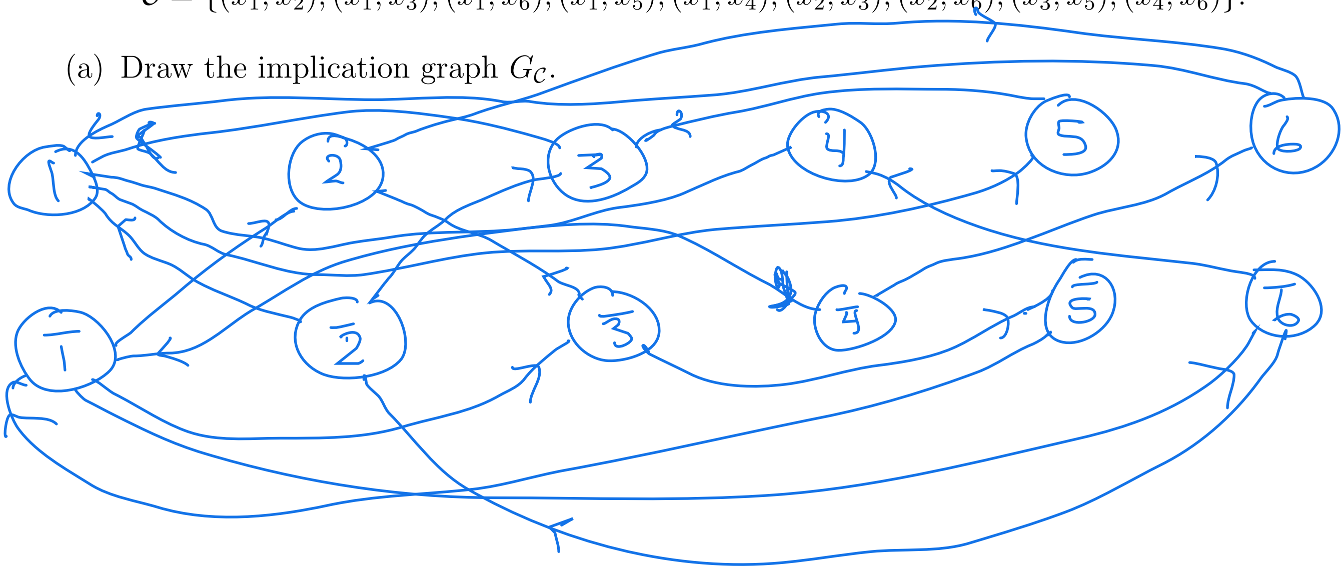
(c) Apply the recurrence from Part b to the graph below in order to calculate $\mathrm{mc}(1, \{2, 3, 4\})$ Show all the necessary computations and use the solutions to compute an optimal path for the salesperson.



LO5. Consider the `2SAT` instance

$$\mathcal{C} = \{(x_1, x_2), (x_1, \overline{x}_3), (x_1, \overline{x}_6), (\overline{x}_1, x_5), (\overline{x}_1, \overline{x}_4), (x_2, x_3), (\overline{x}_2, x_6), (x_3, \overline{x}_5), (x_4, x_6)\}.$$

(a) Draw the implication graph $G_{\mathcal{C}}$.



(b) Perform the `Improved 2SAT` algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced `2SAT` instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable.

$R_{x_1} = \{x_1, x_5, x_3, \overline{x_4}, x_6\}$ is consistent

Satisfying assignment: $\alpha = (x_1 = 1, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 1)$

Note: x_2 can be assigned 1 or 0.

(c) Suppose `2SAT` instance $\mathcal{C}$ is satisfiable and uses 115 variables and 336 clauses. Using the original `2SAT` algorithm, what is the *least* number of queries to a `Reachability` oracle that needs to be made in order to establish $\mathcal{C}$'s satisfiability. In other words, if we make fewer than this number of queries then it is possible that the `2SAT` instance $\mathcal{C}$ may be unsatisfiable. Explain.

115 queries; one per variable in which case all oracle answers evaluate to 0.

LO6. Answer/Solve the following.

(a) Provide the definition of what it means to be a mapping reduction from problem A to problem B.

See Notes

(b) Given the instance (S, t) of `Subset Sum (SS)`, where $S = \{5, \underline{17}, 20, 21, \underline{30}, 34, \underline{41}, 44\}$ and $t = 88$, use the mapping reduction $f : \texttt{SS} \rightarrow \texttt{SP}$ from `SS` to `Set Partition` described in lecture to compute $f(S, t)$.

$t = 88$, $M = \sum_{s \in S} s = 212$

$88 = t < \frac{M}{2} = 106 \Rightarrow$

$f(S,t) = S \cup (M - 2t) = S \cup \{36\}$

(c) Verify that f is valid for input (S, t) from part b in the sense that both (S, t) and $f(S, t)$ are either both positive instances or both negative instances. Make sure your answer is specific to the instances from part b.

(S,t) is a positive instance of SS via subset $A = \{17, 30, 41\}$, while $f(S,t)$ is a positive instance of SP via subsets $A = \{17, 30, 41, J = 36\}$ and $B = \{5, 20, 21, 34, 44\}$ since both sum to 124 and $A \cup B = S' = f(S,t)$.

LO7. An instance of `Additive Inverse Complete (AIC)` is a an array a of n nonzero integers, and the problem is to decide if there is a subarray of a that is **additive-inverse complete**, meaning that if some integer x is in the subarray, then so is $-x$. In other words, there are indices i and j, $0 \le i \le j < n$, for which

$$a[i], a[i+1], \ldots, a[j]$$

is additive-inverse complete. For example, if

$$a = -4, 2, 3, -7, -3, 0, -12, -3, 12, 12, 7, 4, -6$$

then a is a positive instance of `AIC` since for $i = 2$ and $j = 10$ the members of $a[2:10]$ are $3, -7, -3, 0, -12, -3, 12, 12, 7$ which is has the AIC property.

(a) For given instance a of `AIC` describe a certificate in relation to a.

(i, j) is a certificate for a, where $0 \le i \le j < n$.

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance a, and ii) a certificate for a as defined in part a, and decides if the certificate is valid for a.

For each k in $\{i, \ldots, j\}$,
  $x = 0 - a[k]$.
  Found $= 0$.
  For each l in $\{i, \ldots, j\}$,
    If $(a[l] = x)$,
      Found $= 1$
      Break.
  If Found $= 0$, then Return 0.
Return 1.

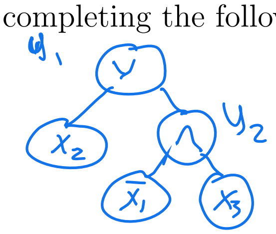
(c) Provide an appropriate size parameter for `AIC`.

$n = |a|$

(d) Use the size parameter from part c to describe the running time of your verifier from part b. Defend your answer.

$O(n^2)$ since the outer For-loop requires $O(n)$ iterations and for each iteration the inner loop requires $O(n)$ iterations.

LO8. Consider the Boolean formula $F = x_2 \vee (\overline{x}_1 \wedge x_3)$. Demonstrate how the **Tseytin transformation** transforms F to an instance of `3SAT`. Do this by completing the following steps.



(a) Draw F's parse tree and label its internal nodes with appropriate y-variables. Then write a Boolean formula that uses both the original and y-variables to assert the satisfiability of F.

$y_1 \wedge (y_1 \leftrightarrow (x_2 \vee y_2)) \wedge (y_2 \leftrightarrow (\bar{x}_1 \wedge x_3))$

(b) Convert the formula from the previous step to a logically-equivalent formula that is in conjunctive normal form (i.e. is an AND of OR's). Do this by using appropriate logical identities: $\leftrightarrow$ to $\rightarrow$, $\rightarrow$ to $\vee$, De Morgan's rule, and Distributivity of $\vee$ over $\wedge$.

$y_1 \wedge (y_1 \rightarrow (x_2 \vee y_2)) \wedge ((x_2 \vee y_2) \rightarrow y_1) \wedge (y_2 \rightarrow (\bar{x}_1 \wedge x_3)) \wedge ((\bar{x}_1 \wedge x_3) \rightarrow y_2) \Longleftrightarrow$

$y_1 \wedge (\bar{y}_1 \vee x_2 \vee y_2) \wedge (\overline{x_2 \vee y_2} \vee y_1) \wedge (\bar{y}_2 \vee (\bar{x}_1 \wedge x_3)) \wedge (\overline{\bar{x}_1 \wedge x_3} \vee y_2) \Longleftrightarrow$

$y_1 \wedge (\bar{y}_1 \vee x_2 \vee y_2) \wedge (\bar{x}_2 \vee y_1) \wedge (\bar{y}_2 \vee y_1) \wedge (\bar{y}_2 \vee \bar{x}_1) \wedge (\bar{y}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_3 \vee y_2)$

(c) Convert the formula from the previous step to an instance of 3SAT, using clause notation, and ensuring that each clause has three literals.

$\{(y_1, y_1, y_1), (\bar{y}_1, x_2, y_2), (\bar{x}_2, y_1, y_1), (\bar{y}_2, y_1, y_1), (\bar{y}_2, \bar{x}_1, \bar{x}_1), (\bar{y}_2, x_3, x_3), (x_1, \bar{x}_3, y_2)\}$