CECS 528, LO9 Assessment, April 17th, 2024, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each solution on a separate sheet of paper.

Problems

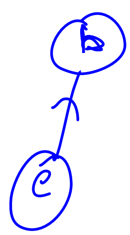
LO5. For the weighted graph with edges

$$(a, e, 3), (b, e, 1), (c, d, 2), (c, e, 5), (d, e, 4), (e, f, 6),$$

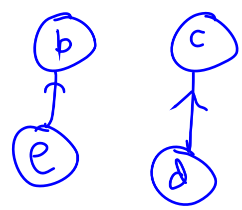
Show how the membership-tree forest (not the Kruskal forest!) changes when processing each edge in the Kruskal sorted order when performing Kruskal's algorithm. When merging two trees, use the convention that the root of the merged tree should be the one having *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are merged, then the merged tree should have root a.

Solution.

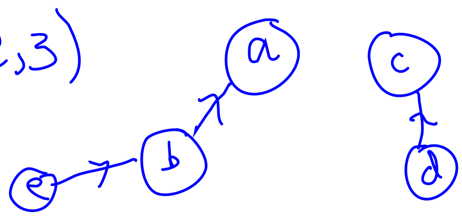
1. (b,e,1)



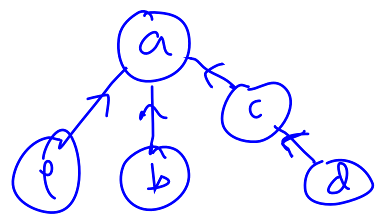
2. (c,d,2)



3. (a,e,3)

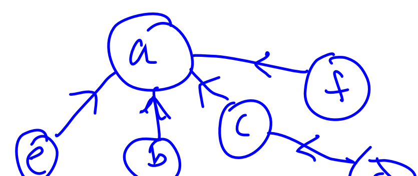


4. (d,e,4)



5. (c,e,5) No change

6. (e,f,6)



LO6. Answer the following with regards to a correctness-proof outline for Dijkstra's algorithm.

(a) In relation to Dijkstra's algorithm, provide a definition for what it means to be i) an i-neighboring path from source s to an external vertex v, and ii) the i-neighboring distance $d_i(s, v)$ from source s to external vertex v. Hint: at this point in the algorithm i nodes have been added to the DDT.

Solution. An i-neighboring path from source s to an external vertex v is a path from s to v that uses exactly one edge that is not in T_i, the Dijkstra Distance Tree after Round i of Dijkstra's algorithm. The i-neighboring distance from source s to an external vertex v is equal to the cost of the minimum-cost i-neighboring path from s to v.

(b) Using the definitions from part a, describe the greedy choice that is made in each round of Dijkstra's algorithm.

Solution. For Round $i+1$ the vertex chosen by Dijkstra's algorithm is that vertex that is external to T_i and has the minimum i-neighboring distance $d_i(s, v)$ from source s among all external vertices.

(c) If vertex v is chosen by Dijkstra in Round $i+1$, use part b to prove that $d(s, v) = d_i(s, v)$. Hint: if i-neighboring path P from s to v has cost $d_i(s, v)$ and Q is any other path from s to v, explain why $\text{cost}(Q) \geq d_i(s, v)$.

Solution. Any path P from s to v must use at least one edge that is external to T_i. Thus, P must have an i-neighboring subpath from s to v. Moreover, the cost of this path must be at least as much as the cost of minimum i-neighboring path from s to v which was used to determine $d_i(s, v)$. Hence $\text{cost}(P) \geq d_i(s, v)$ hence the minimum-cost i-neighboring path from s to v has a cost that equals $d(s, v)$, the actual distance from s to v. This is true, since its cost does not exceed the cost of any path P from s to v.

LO7. Answer the following.

(a) Provide the dynamic-programming recurrence for computing the maximum-cost path, denoted $\text{mc}(u, v)$, from a vertex u to a vertex v in a directed acyclic graph (DAG) $G = (V, E, c)$, where $c(x, y)$ gives the cost of edge $e = (x, y)$, for each $e \in E$. The recurrence should allow one to compute the maximum costs from a single source to all other vertices in a linear number of steps. Hint: step backward from v.

Solution.

See Solution to last week's LO7 Solution

(b) Draw the vertices of the following DAG G in a linear left-to-right manner so that the vertices are topologically sorted, meaning, if (u, v) is an edge of G, then u appears to the left of v. The vertices of G are a-h, while the weighted edges of G are

$$(a, b, 6), (a, e, 6), (a, f, 3), (b, c, 7), (b, g, 4), (c, d, 2), (c, g, 2), (c, h, 4), (d, h, 8), (e, b, 6), (e, f, 2),$$

$$(f, b, 3), (f, c, 3), (f, g, 3), (g, d, 4), (g, h, 2).$$

Solution.

(See Solution to last week's LO7 Problem. Same graph, different weights)

(c) Starting from left to right in topological order, use the recurrence to compute

$$\mathrm{mc}(a, a), \ldots, \mathrm{mc}(a, h).$$

Solution.

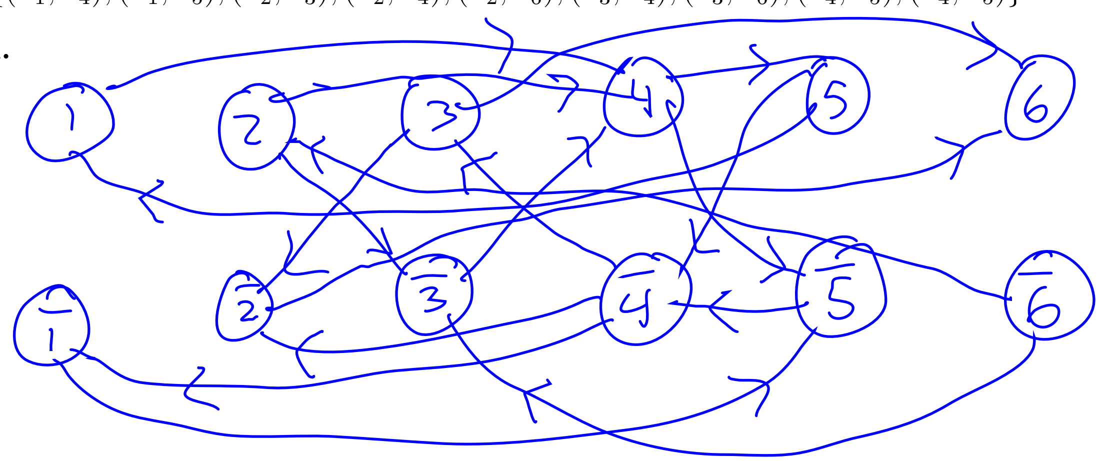
See Last week's Solution to LO7 Problem (Same graph different weights)

LO8. Do/answer the following.

(a) Draw the implication graph $G_{\mathcal{C}}$ associated with the `2SAT` instance

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

Solution.



(b) Apply the improved `2SAT` algorithm to obtain a satisfying assignment for $\mathcal{C}$. When deciding on the next reachability set R_l to compute, follow the literal order $l = x_1, \overline{x}_1, \ldots, x_4, \overline{x}_4$. For each consistent reachability set encountered, provide the partial assignment α_{R_l} associated with R_l and draw the reduced implication graph before continuing to the next reachability set. Note: do *not* compute the reachability set for a literal that has already been assigned a truth value. Provide a final assignment α and verify that it satisfies all six clauses.

Solution. $R_{x_1} = \{x_1, x_4, \overline{x}_5, \overline{x}_4, \overline{x}_2, x_6, \overline{x}_1\}$ is inconsistent

$R_{\overline{x}_1} = \{\overline{x}_1, \overline{x}_5, \overline{x}_4, \overline{x}_2, x_3, x_6\}$

satisfying assignment $\alpha_{R_{\overline{x}_1}} = (x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 1)$

(c) If an instance $\mathcal{C}$ has 336 variables and 2027 clauses, then what is the worst-case number of queries that must be made to the `Reachability` oracle in order to confirm that $\mathcal{C}$ is unsatisfiable? Explain.

672

Solution. We need at most $2 \times 336 =$ [redacted] queries in the worst case since the final variable (and its negation) that is considered in terms of possibly being in an inconsistent cycle with its negation could be the only one that is in such a cycle. Moreover, we need two oracle queries, `reachable`$(x, \overline{x})$ and `reachable`$(\overline{x}, x)$, to check for an inconsistent cycle.

LO9. Answer the following.

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

Solution. See Turing and Mapping Reducibility Lecture.

(b) For the mapping reduction f : `Subset Sum` $\to$ `Set Partition`, determine $f(S, t)$ for `Subset Sum` instance ($S = \{8, 11, 13, 17, 19, 22, 26, 29\}, t = 58$). Show work.

Solution. $M = 145$ Since $58 < M/2$, $J = 145 - 2(58) = 29$

So, $f(S,t) = S \cup \{J = 29\}$

(c) Verify that both (S, t) and $f(S, t)$ are either both positive instances or both negative instances of their respective decision problems. If both are positive, then provide valid certificates for each. Otherwise, explain why neither has a valid certificate.

Solution. $A = \{8, 11, 17, 22\}$ is a solution for (S,t) Since $8 + 11 + 17 + 22 = 58$.

Also, $A' = \{8, 11, 17, 22, \overset{J}{29}\}$, $B' = \{13, 19, 26, 29\}$ is a set partion for $f(S,t)$, since both sum to 87, $A' \cup B' = S \cup \{J = 29\}$, and $A' \cap B' = \emptyset$

Assuming the two 29's are distinct