CECS 528, Learning Outcome Assessment 10, Yellow, Fall 2023, Dr. Ebert

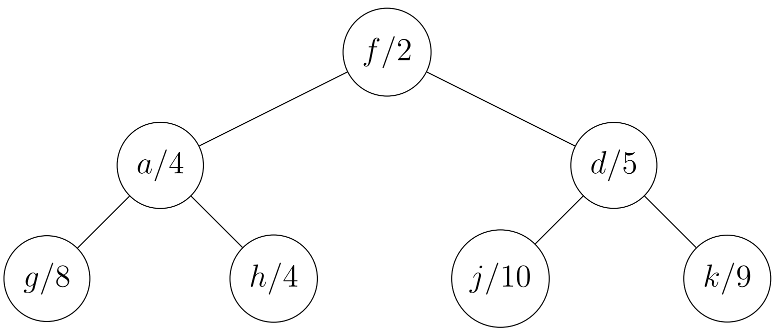
NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each solution on a separate sheet of paper.

Problems

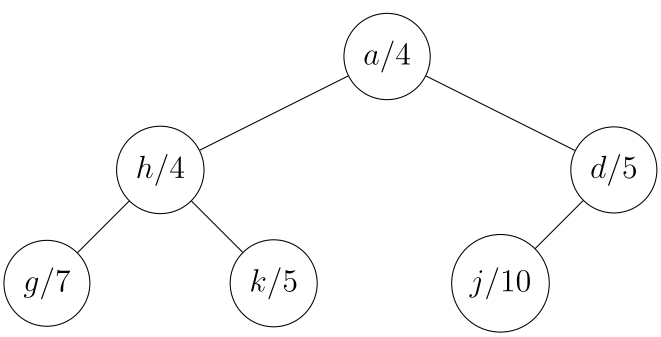
LO6. The tree below shows the state of the binary min-heap at the beginning of some round of Dijkstra's algorithm, applied to some weighted graph G. If G has directed edges

$$(a, f, 1), (d, g, 2), (f, g, 5), (f, h, 2), (f, k, 3),$$

then draw a plausible state of the heap at the end of the round.



Solution.



LO7. Do the following. Note: correctly solving this problem counts for passing LO7.

(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 $\text{mc}(i, A)$. In words, what does $\text{mc}(i, A)$ equal? Hint: do *not* write the recurrence (see Part b). Hint: we call it "Runaway TSP" because the salesperson does *not* return home.

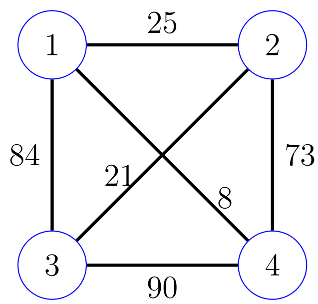
Solution. $\text{mc}(i, A)$ equals the cost of the minimum-cost simple path that starts at i and visits every vertex in A.

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

Solution. We have

$$\text{mc}(i, A) = \begin{cases} 0 & \text{if } A = \emptyset \\ C_{ij} & \text{if } A = \{j\} \\ \min\limits_{j \in A}(C_{ij} + \text{mc}(j, A - \{j\}) & \text{otherwise} \end{cases}$$

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



Solution. Start with $\text{mc}(1, \{2, 3, 4\})$ and proceed to compute other mc values as needed.

$$\text{mc}(1, \{2, 3, 4\}) = \min(25 + \text{mc}(2, \{3, 4\}), 84 + \text{mc}(3, \{2, 4\}), 8 + \text{mc}(4, \{2, 3\})).$$

$$\text{mc}(2, \{3, 4\}) = \min(21 + \text{mc}(3, \{4\}), 73 + \text{mc}(4, \{3\})) = \min(21 + 90, 73 + 90) = 111.$$

$$\text{mc}(3, \{2, 4\}) = \min(21 + \text{mc}(2, \{4\}), 90 + \text{mc}(4, \{2\})) = \min(21 + 73, 90 + 73) = 94.$$

$$\text{mc}(4, \{2, 3\}) = \min(73 + \text{mc}(2, \{3\}), 90 + \text{mc}(3, \{2\})) = \min(73 + 21, 90 + 21) = 94.$$

Therefore,

$$\text{mc}(1, \{2, 3, 4\}) = \min(25 + \text{mc}(2, \{3, 4\}), 84 + \text{mc}(3, \{2, 4\}), 8 + \text{mc}(4, \{2, 3\})) =$$

$$\min(25 + 111, 84 + 94, 8 + 94) = 102.$$

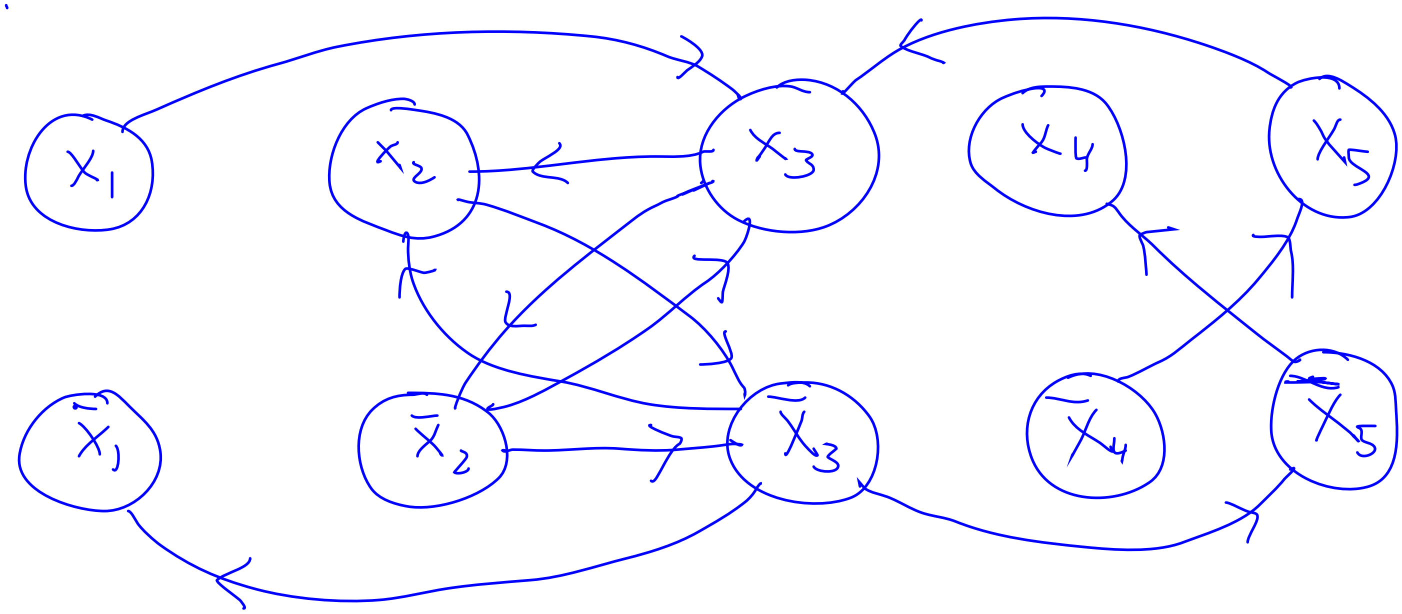
This gives the optimal path $P = 1, 4, 2, 3$.

LO8. Do/answer the following.

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

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

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 $l = x_1, \overline{x}_1, \ldots, x_5, \overline{x}_5$. 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 the clauses.

Solution. $R_{x_1} = \{x_1, \overline{x}_1, \ldots, x_4, \overline{x}_5\}$ is inconsistent, while $R_{\overline{x}_1} = \{\overline{x}_1\}$ which is consistent and so $\alpha_{R_{\overline{x}_1}} = (x_1 = 0)$. Now remove x_1 and $\overline{x}_1$ from the implication graph including all edges incident with these vertices. Then $R_{x_2} = \{x_2, \overline{x}_3, x_4, \overline{x}_5\}$ is consistent, and so $\alpha_{R_{\overline{x}_2}} = (x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0)$. Final assignment:

$$\alpha = \alpha_{R_{\overline{x}_1}} \cup \alpha_{R_{x_2}} = (x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0).$$

(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.

Solution. The least number of queries is 115 in which case the `Reachability` oracle would answer no for each `reachable`$(G_{\mathcal{C}}, x_i, \overline{x}_i)$ query, $i = 1, \ldots, 115$.

LO9. Answer the following questions?

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

Solution. See Definition 2.1 of the Mapping Reducibility lecture.

(b) As part of her network security project, Julie is working with a simple graph H that has 145 vertices and 372 edges. She needs to know whether or not H has a Hamilton path. Her colleague Simon has implemented the Python function

```
Boolean has_LPath(Graph G, int k);
```

that decides if the input graph G has a simple path of length k. Explain how Julie can use Simon's function to answer her question. If used properly, why will Simon's function return the answer that Julie seeks.

Solution. Julie should call Simon's `has_LPath` function with inputs H and $k = 144$ since H has a Hamilton path iff H has a simple path of length $|V| - 1 = 144$.

LO10. An instance of the `Feedback Arc Set (FAS)` decision problem is a pair (G, k) where $G = (V, E)$ is a directed graph and k is a nonnegative integer. The problem is to decide if there is a subset E' of k edges that can be removed from G so that $G - E'$ is an acyclic graph.

(a) For a given instance (G, k) of `FAS`, describe a certificate in relation to (G, k).

Solution. A certificate is a subset E' of k edges.

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance (G, k), ii) a certificate for (G, k) as defined in part a, and decides if the certificate is valid for (G, k). Do this by making use of a reachability oracle via function $\texttt{reachable}(H, a, b)$ which returns true iff vertex b is reachable from vertex a in graph H. Hint: for each vertex $v \in V$, consider the set $N^+(v) = \{u | (u, v) \in E\}$.

Solution.

For each $v \in V$,
  For each $u \in N^+(v)$,
    If $(u, v) \notin E' \wedge \texttt{reachable}(G - E', v, u)$, then return 0.
Return 1.

(c) Provide size parameters that may be used to measure the size of an instance (G, k) of `FAS`.

Solution. $m = |E|$ and $n = |V|$

(d) Use the size parameters from part c to describe the running time of your verifier from part b. Assume that a single call to function `reachable` requires a linear number of steps with respect to the size of the input graph. Defend your answer in relation to the algorithm you provided for the verifier.

Solution. A total of $O(n^2)$ iterations must be taken within the nested `for`-loops, and $O(m+n)$ steps are required to answer the `reachable` query, for a total of $O(n^2(n+m)) = O(n^2 m)$ steps, which is cubic in m and n.