CECS 528, Exam 2, Fall 2024, Dr. Ebert

Directions: This exam has nine different problems: one problem for each of LO's 1-7 and two advanced problems.

- **SOLVE AT MOST SIX PROBLEMS.**
- Use a **SINGLE SHEET OF PAPER** per problem. Write **NAME** and **PROBLEM NUMBER** on sheet.
- Do **NOT** write solutions to different problems on the same sheet.
- Example 1: LO7 has three parts (a,b,c). All parts should be written on the **SAME SHEET**.
- Example 2: Jason chose to solve **five problems**. Therefore, Jason will submit **FIVE ANSWER SHEETS**.
- A 20% deduction in points will be applied to each solution that does not follow the above guidelines.

Unit 2 LO Problems (25 pts each)

LO4. Answer the following.

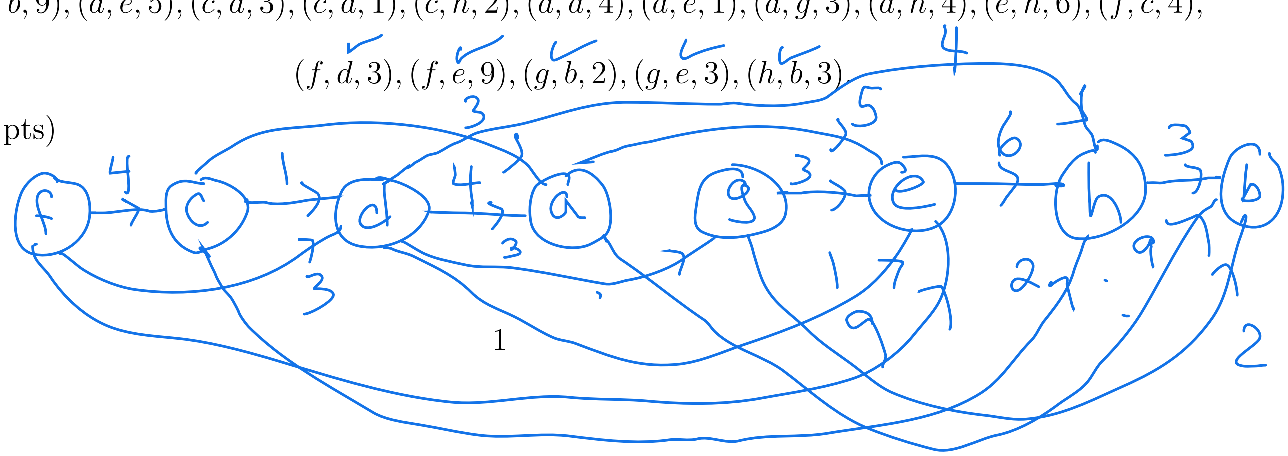
(a) Provide the dynamic-programming recurrence for computing the distance $\mathrm{d}(u, v)$, from a single-source vertex u to a vertex v in a directed acyclic graph (DAG) $G = (V, E, c)$, where $c(x, y)$ gives the cost of edge $e = (x, y)$, for each $e \in E$. Hint: step *backward* from v, rather than forward from u. (7 pts)

See Notes

(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

$$(a, b, 9), (a, e, 5), (c, a, 3), (c, d, 1), (c, h, 2), (d, a, 4), (d, e, 1), (d, g, 3), (d, h, 4), (e, h, 6), (f, c, 4),$$
$$(f, d, 3), (f, e, 9), (g, b, 2), (g, e, 3), (h, b, 3)$$

(8 pts)



(c) Starting from left to right in topological order, use the recurrence to compute the distance from a to every vertex. Show the computations. (10 pts)

$d(f,f) = 0 \quad d(f,c) = 4 \quad d(f,d) = \min(\underset{f}{0+3}, \underset{c}{4+1}) = 3$

$d(f,a) = \min(\underset{c}{4+3}, \underset{d}{3+4}) = 7 \quad d(f,g) = 3+3 = 6$

$d(f,e) = \min(\underset{a}{7+5}, \underset{f}{0+9}, \underset{d}{3+1}, \underset{g}{6+3}) = 4$

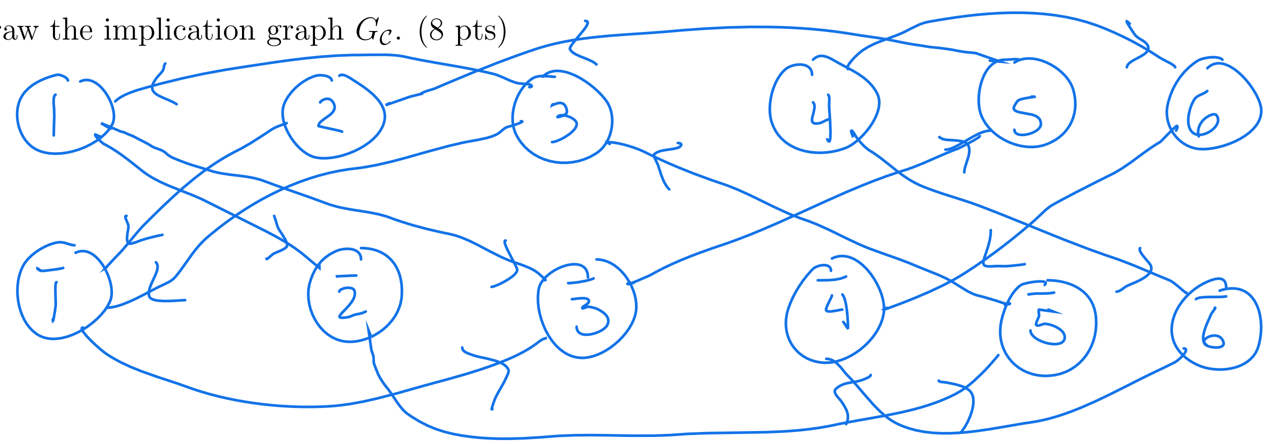
$d(f,h) = \min(\underset{c}{4+2}, \underset{d}{3+4}, \underset{e}{4+6}) = 6$

$d(f,b) = \min(\underset{a}{7+9}, \underset{g}{6+2}, \underset{h}{7+3}) = 8$

LO5. Consider the 2SAT instance

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

(a) Draw the implication graph $G_{\mathcal{C}}$. (8 pts)



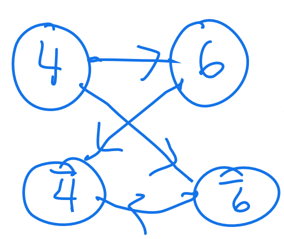
(b) Perform the Improved 2SAT algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced 2SAT instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable. (10 pts)

$R_{x_1} = \{x_1, \overline{x}_2, \overline{x}_5, x_3, \overline{x}_1, \overline{x}_3, x_5, x_2\}$ Inconsistent

$R_{\overline{x}_1} = \{\overline{x}_1, \overline{x}_3, x_5, x_2\}$ is consistent.

$\alpha_{R\overline{x}_1} = (x_1 = 0, x_2 = 1, x_3 = 0, x_5 = 1\}$

Reduced Graph:



$\Longrightarrow$ $R_{x_4} = \{x_4, x_6, \overline{x}_4, \overline{x}_6\}$ is inconsistent.

$R_{\overline{x}_4} = \{\overline{x}_4\}$ is consistent.

$\alpha_{\overline{x}_4} = \{x_4 = 0\}$

Satisfying Assignment:

$\alpha = \alpha_{R\overline{x}_1} \cup \alpha_{R\overline{x}_4} = (x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0, x_5 = 1\}$

Note: x_6 can be assigned either 0 or 1.

(c) Satisfiable instance $\mathcal{C}$ of `2SAT` has six variables and nine clauses. Moreover, both $\alpha_1 = (x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 0, x_6 = 0)$ and $\alpha_2 = (x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 0, x_6 = 1)$ are the only two assignments that satisfy $\mathcal{C}$. How many `Reachability`-oracle queries are made during the `2SAT` algorithm when $\mathcal{C}$ serves as input. Hint: you may assume that, for any variable x, the query reachable$(G_{\mathcal{C}}, x, \overline{x})$ is performed first. Explain. (7 pts)

At least $8 = 6 + 2$ queries needed. If a satisfying assign. has $x_i = 1$, then $R(G_C, x_i, \bar{x}_i) = 0$

	$R(G_C, x, \bar{x})$	$R(G_C, \bar{x}, x)$
x_1	0	n/a
x_2	1	0
x_3	0	n/a
x_4	0	n/a
x_5	1	0
x_6	0	n/a

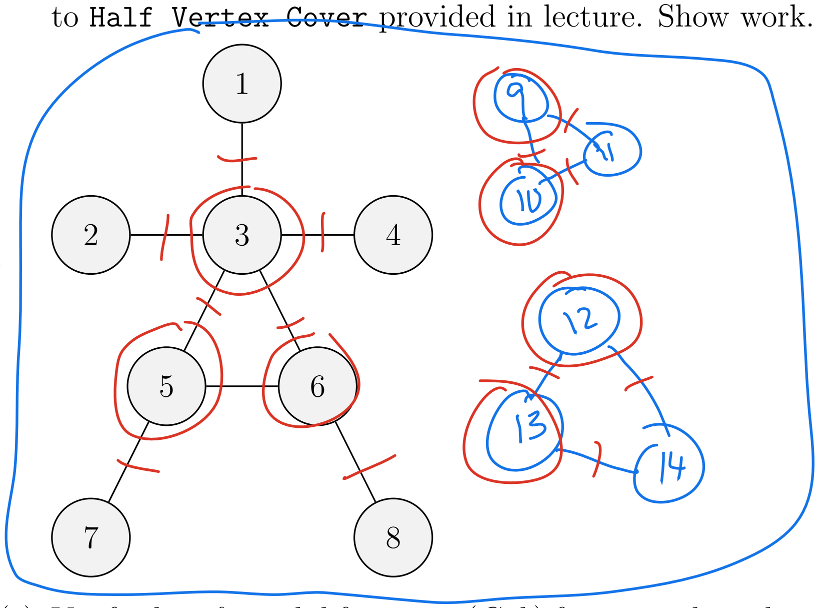
LO6. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from problem A to problem B. (5 pts)

See Notes

(b) The simple graph $G = (V, E)$ shown below along with $k = 3$ is an instance of the `Vertex Cover (VC)` decision problem. Draw $f(G, k)$, where f is the mapping reduction from `VC` to `Half Vertex Cover` provided in lecture. Show work. (10 pts)

$f(G,k) =$



Add $n - 2k = 8 - 6 = 2$ Δs.

(c) Verify that f is valid for input (G, k) from part b in the sense that both (G, k) and $f(G, k)$ are either both positive instances or both negative instances. Make sure your answer is specific to the instances from part b. (10 pts)

(G,k) is positive for VC via cover $\{3,5,6\}$ while $f(G,k)$ is positive for HVC via half-cover $\{3,5,6,9,10,12,13\}$ which has size $7 = \frac{14}{2}$.

LO7. An instance of the `Perfect Matching` decision problem is a bipartite graph $G = (V_1, V_2, E)$, where $|V_1| = |V_2| = n$ for some integer $n \geq 1$, and every $e \in E$ is incident with one vertex in V_1 and one vertex in V_2. The problem is to decide if G has a matching of size n. We now establish that `Perfect Matching` is an `NP` problem.

(a) For a given instance $G = (V_1, V_2, E)$ of `Perfect Matching`, describe a certificate in relation to $G = (V_1, V_2, E)$. (7 pts)

$M \subseteq E$ is a subset of n edges.

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance $G = (V_1, V_2, E)$ of `Perfect Matching`, ii) a certificate for $G = (V_1, V_2, E)$ as defined in part a, and decides if the certificate is valid for $G = (V_1, V_2, E)$. (11 pts)

Let T be an empty table of size $2n$.
For each $e = (u, v) \in M$
  If $(u \in T \vee v \in T)$, Return 0.
  Insert u and v into T.
Return 1.

// u or v is incident with more than one edge in M

(c) Use the size parameters $m = |E|$ and $n = |V|$ to describe the running time of your verifier from part b. Defend your answer in relation to the algorithm you provided for the verifier. (7 pts)

Building T: $O(n)$
The For-loop requires $O(n)$ iterations with $O(1)$ steps (2 table lookups and insertions) per iteration. $\therefore$ $O(n)$ total.

Advanced Problems

A1. Answer the following.

(a) Suppose $A \leq_T B$ via algorithm $\mathcal{A}_{AB}$ and $B \leq_T C$ via algorithm $\mathcal{A}_{BC}$. Describe how $\mathcal{A}_{AB}$ and $\mathcal{A}_{BC}$ can be used to develop an algorithm $\mathcal{A}_{AC}$ that establishes $A \leq_T C$. (20 pts)

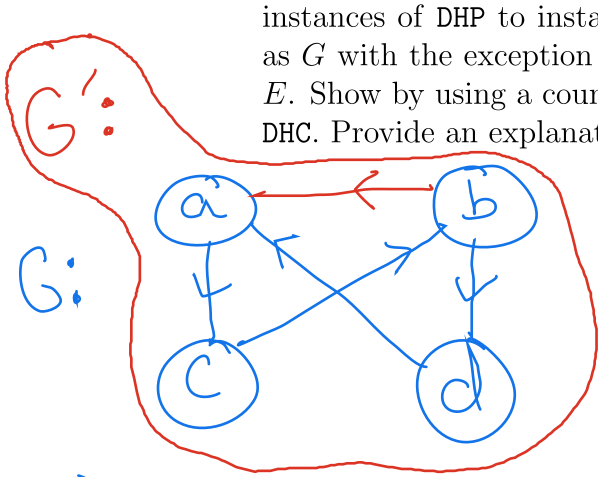
See Solution to Exercise 4

(b) Suppose further that $\mathcal{A}_{AB}$ requires $O(n^4)$ steps to compute an input a of size n (here n is the size parameter for problem A). Similarly, $\mathcal{A}_{BC}$ requires $O(m^3)$ steps to compute an input b of size m (here m is the size parameter for problem B). Based on your answer to part a, how many steps would be required for $\mathcal{A}_{AC}$ to compute an instance a of size n? Explain. (15 pts)

A_{AC} requires $O(n^4 \cdot (n^4)^3) = O(n^{16})$ steps. This is because, when A queries the B oracle, the size of the query word is $O(n^4)$. Then to answer the query algorithm A_{BC} is run which requires $O((n^4)^3) = O(n^{12})$ steps. Finally at most $O(n^4)$ queries are made for a total of $O(n^{16})$ steps.

A2. Answer the following.

(a) The `Directed Hamilton Path (DHP)` decision problem consists of a directed graph $G = (V, E)$ and two vertices $a, b \in V$, and the problem is to decide if there is a Hamilton path in G that starts at a and ends at b. Similarly, the `Directed Hamilton Cycle (DHC)` decision problem consists of a directed graph $G = (V, E)$, the problem is to decide if there is a Hamilton cycle in G, i.e. a cycle that visits every vertex in V once and then returns to the starting vertex. Consider the following function/map f : `DHP` $\to$ `DHC` that maps instances of `DHP` to instances of `DHC`, where $f(G, a, b) = G'$, where G' is the same graph as G with the exception that the edge (b, a) is added to G if it does not already exist in E. Show by using a counterexample that f is *not* a valid mapping reduction from `DHP` to `DHC`. Provide an explanation of why your example is in fact a counterexample. (20 pts)



(G, a, b) is negative for DHP but G' is positive for DHC via cycle a, c, b, d, a. ∴ f is not a valid mapping reduction.

(b) Modify f so that it becomes a valid mapping reduction. Give a general proof (i.e. it is insufficient to only provide examples) that your modified f is valid. (15 pts)

$f(G, a, b) = G'$ where G' is the same graph as G but with an extra vertex V added along with edges (b, V), and (V, a). As before, if G has a DHP from a to b, then G' has a DHC by adding edges (b, V), and (V, a). Conversely, if G' has a DHC, then it must include the subpath b, V, a, and the remainder of the cycle is a DHP from a to b.

1 Unit 1 LO Problems (0 pts each)

LO1. Solve each of the following problems.

a. Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 4T(n/2) + n^{\log_2 6}$.

b. Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 4T(n/2) + 3n,$$

Then $T(n) = \mathrm{O}(n^2)$.

LO2. Solve the following problems.

a. State two fundamental properties of the n th roots of unity that play critical roles in the correctness of the FFT algorithm.

See Notes

b. Draw the recursion tree that results when applying Mergesort to the array

$$a = 4, 16, 8, 20, 5, 6, 9, 3, 10, 17,$$

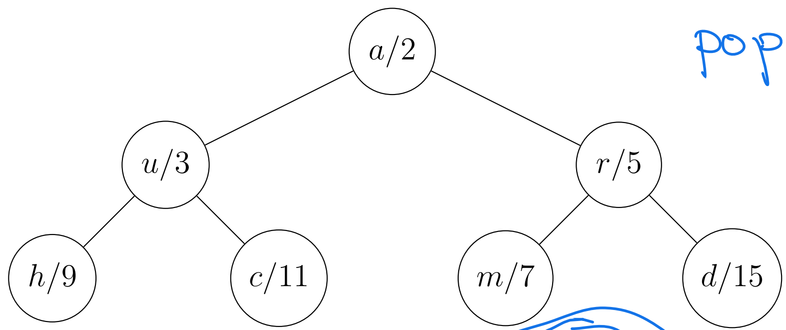
Label each node with the subproblem to be solved at that point of the recursion. Assume arrays of size 1 and 2 are base cases. Assume that odd-sized arrays are split so that the left subproblem has one more integer than the right. Next to each node, write the solution to its associated subproblem.

LO3. Solve each of the following problems.

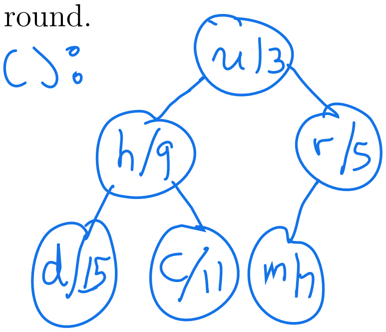
a. The tree below shows the state of the binary min-heap at the beginning of some round of Prim's algorithm, applied to some weighted graph G. If G has edges

$$(h, u, 2), (a, m, 4), (c, a, 3), (a, p, 2), (a, u, 5), (p, a, 2),$$

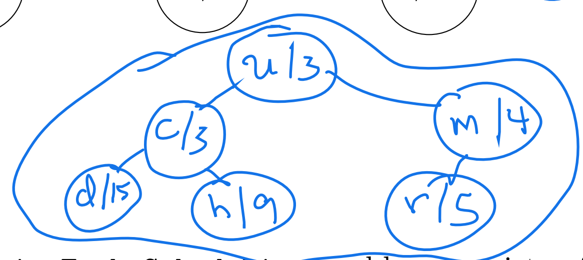
then draw a plausible state of the heap at the end of the round.



pop():



Increase priorities: m: 7 ⇒ 4
c: 11 ⇒ 3



b. An instance of the `Unit Task Scheduling` problem consists of the tasks shown in the table below.

Task	a	b	c	d	e	f	g	h	i	j	k
Deadline	4	8	6	6	4	2	3	2	8	7	1
Profit	50	80	30	30	70	30	70	50	80	20	60

State the greedy choice that is being made in each round of the algorithm and use it to obtain an optimal schedule. Show work.

Choose the task of next highest profit and schedule it as close to its deadline as possible (but not after its deadline), or skip it if there are no openings.

Round		Location
1	b	8
2	i	7
3	e	4
4	g	3
5	k	1
6	a	2
7	h	n/a
8	c	6
9	d	5
10	f	n/a
11	j	n/a

Optimal schedule:

k	a	g	e	d	c	i	b
1	2	3	4	5	6	7	8

Total profit = $\boxed{470}$