

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

Problems

LO6. The following pertains to a correctness-proof outline for the Unit Task Scheduling (UTS) algorithm. Let $S = (a_1, t_1), \dots, (a_m, t_m)$ represent the tasks that were selected by the algorithm for scheduling, where a_i is the task, and t_i is the time that it is scheduled to be completed, $i = 1, \dots, m$. Moreover, assume that these tasks are ordered in the same order for which they appear in the sorted order. Let S_{opt} be an optimal schedule which also consists of task-time pairs. Let k be the first integer for which $(a_1, t_1), \dots, (a_{k-1}, t_{k-1})$ are in S_{opt} , but $(a_k, t_k) \notin S_{\text{opt}}$ because a_k is scheduled by S_{opt} , but at time $t \neq t_k$.

- (a) Explain why $t < t_k$. Assume that $t > t_k$ and explain why this creates a contradiction.
- (b) Assume that S_{opt} has scheduled some task a at time t_k . Explain why

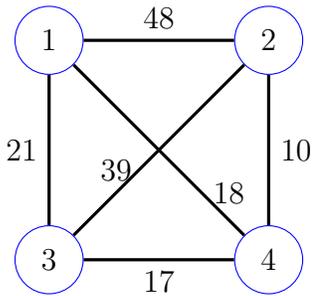
$$\hat{S}_{\text{opt}} = S_{\text{opt}} - \{(a_k, t), (a, t_k)\} + \{(a_k, t_k), (a, t)\}$$

is a valid schedule. In words, the new schedule swaps schedule times for a_k and a . Explain why this does not create a scheduling problem for either task.

- (c) Continuing in this manner we eventually arrive at an optimal schedule S_{opt} for which $S \subseteq S_{\text{opt}}$. Moreover, explain why it is not possible for S_{opt} to possess a task-time pair (a, t) that is *not* a member of S . Assuming it did have such a pair, what contradiction arises?

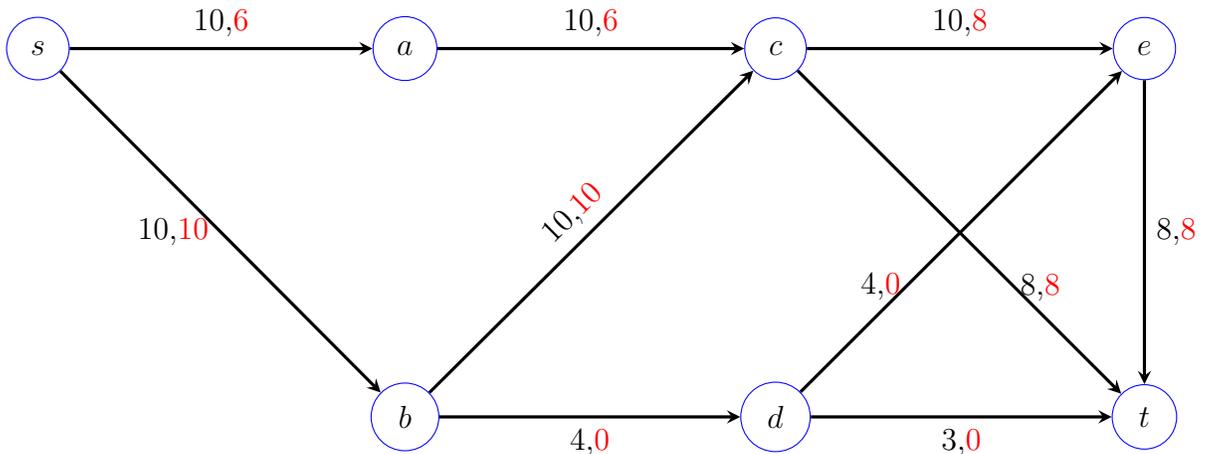
LO7. Do the following.

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



LO8. A flow f (2nd value listed on each edge) has been placed in the network G below. Note: correctly solving this problem counts for passing LO8.

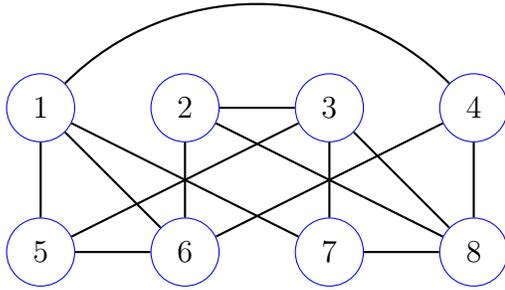
- (a) Draw the residual network G_f and use it to determine an augmenting path P . Highlight path P in the network so that it is clearly visible. (10 pts)



- (b) In the original network, cross out any flow value that changed, and replace it with its updated value from $f_2 = \Delta(f, P)$. (7 pts)
- (c) What one query can be made to a **Reachability** oracle to determine if f_2 is a maximum flow for G ? Hint: three inputs are needed for the **reachable** query function. Clearly define each of them. (8 pts)

LO9. Answer the following.

- (a) Provide the definition of what it means to be a mapping reduction from problem A to problem B .
- (b) Suppose $(G, k = 3)$ is an instance of the **Vertex Cover** decision problem, where G is drawn below. Draw $f(G, k)$, where f is the mapping reduction from **Vertex Cover** to the **Half Vertex Cover** decision problem.



- (c) Verify that f is valid for input (G, k) in the sense that both (G, k) and $f(G)$ are either both positive instances or both negative instances of their respective problems. Defend your answer. Hint: it takes two vertices to cover each edge of a triangle in a graph.

LO10. Do the following.

- (a) An instance of **Set Cover** is a triple (\mathcal{S}, m, k) , where $\mathcal{S} = \{S_1, \dots, S_n\}$ is a collection of n subsets, where $S_i \subseteq \{1, \dots, m\}$, for each $i = 1, \dots, n$, and a nonnegative integer k . The problem is to decide if there are k subsets S_{i_1}, \dots, S_{i_k} for which

$$S_{i_1} \cup \dots \cup S_{i_k} = \{1, \dots, m\}.$$

Verify that (\mathcal{S}, m, k) is a positive instance of **Set Cover**, where $m = 9$, $k = 4$, and

$$\mathcal{S} = \{\{1, 3, 5, 8\}, \{3, 7, 9\}, \{2, 4, 5\}, \{2, 6, 7\}, \{6\}, \{2, 4, 7, 9\}, \{1, 3, 7\}, \{4, 5\}\}.$$

- (b) For a given instance (\mathcal{S}, m, k) of **Set Cover** describe a certificate in relation to (\mathcal{S}, m, k) .
- (c) Provide a semi-formal verifier algorithm that takes as input i) an instance (\mathcal{S}, m, k) of **Set Cover**, ii) a certificate for (\mathcal{S}, m, k) as defined in part a, and decides if the certificate is valid for (B, k) .
- (d) Provide size parameters that may be used to measure the size of an instance of **Set Cover**.
- (e) Use the size parameters from part d to describe the running time of your verifier from part c. Defend your answer in relation to the algorithm you provided for the verifier.