

Module 2 Graded Quiz

Due Sep 10, 2021 at 11:59pm **Points** 10

Questions 10

Available Aug 28, 2021 at 12am - Jan 28 at 11:59pm 5 months

Time Limit 120 Minutes

Attempt History

	Attempt	Time	Score
LATEST	<u>Attempt 1</u>	35 minutes	10 out of 10

Score for this quiz: **10** out of 10

Submitted Sep 9, 2021 at 9:18pm

This attempt took 35 minutes.

Question 1

1 / 1 pts

In the proof of optimality for the greedy algorithm in interval scheduling, we picked r to be the maximum for which the greedy and optimal solutions were identical choices. Why was this important in the overall proof?



The two methods must agree on some number of jobs, so we are giving a name to that number.



Because it can be done without loss of generality.



None of the other answers is correct.

Correct!

Because a contradiction about r is reached from a substitution-type argument (substituting a job from a greedy schedule into an optimal schedule).

Question 2**1 / 1 pts**

In the proof of Dijkstra's algorithm's being optimal, the base case said that " $|S| = 1$ is trivial", where S is the set of explored nodes that are determined from the source node to each explored node. Why is it trivial?



Because the base case is always one edge, and the distance between any two nodes connected by an edge is length of the edge.



Because a graph with a single vertex can contain no paths



Because a graph with a single edge has exactly one path.



If the size of S is 1, then S must only include the source node. The distance from the source node to itself is always 0, which must be optimal since it is the only possible path.

Correct!**Question 3****1 / 1 pts**

What is the best asymptotic run-time of the greedy algorithm for Interval Partitioning, if there are n intervals?

☐ $O(n)$.

Correct!☐ $O(n^2 \log n)$ ☐ $O(n^2)$ ☒ $O(n \log n)$ **Question 4****1 / 1 pts**

Informally, in the minimizing lateness problem, why does sorting jobs in ascending order of processing time not allow for a greedy-type approach to find an optimal solution?

☐ It is not optimal only there are jobs with different deadlines.☐

In fact, picking shorter jobs is optimal if each job can be completed within the minimum deadline of all jobs.

Correct!☒

Because longer jobs with earlier deadlines are possible, and delaying them by picking shorter jobs makes the solution less optimal.

☐ Because shorter jobs typically have later deadlines.**Question 5****1 / 1 pts**

Which of the following accurately describes a high-level sketch of the proof of optimality for the farthest-in-the-future algorithm for offline caching?

Correct!

Use induction to show that for any given request, a special property holds in all cases.



Use induction to show that each requested object is the same as the previous object, i.e. invariant, beginning with request number 2.



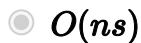
Use a "proof by cases" technique to show that a certain property remains invariant over a single request.



Use induction to show that a certain property remains invariant over a single request.

Question 6**1 / 1 pts**

Suppose that if an item is evicted in the optimal offline caching problem, you store this eviction to a file (for logging purposes); suppose this takes $O(s)$ time for each eviction. If you run the farthest-in-future algorithm with n requests, what is the worst-case run-time for this algorithm variant?

Correct!

It is impossible to know without knowing the number of evictions that will be performed.



$O(s\sqrt{n})$.



$O(n + s)$

Question 7**1 / 1 pts**

Do Kruskal's and Prim's algorithm find an MST where edges can have negative weight?

Correct!

- ☒ Both do.
- ☐ Neither do.
- ☐ Prim's algorithm does, Kruskal's algorithm does not.
- ☐ Kruskal's algorithm does, Prim's algorithm does not.

Question 8**1 / 1 pts**

Does Dijkstra's algorithm work when we allow for edges to have negative weight?

Correct!

- ☒ It may not work.
- ☐ It never works.
- ☐ It always works.

Question 9**1 / 1 pts**

Which of the following descriptions best embodies the paradigm of greedy algorithms?

Correct!

Solve the given problem via a series of locally optimal decisions and never revisit a decision.



Break the given problem down to non-overlapping subproblems, solve these subproblems and combine them to obtain a global solution.



Break the given problem down to overlapping subproblems, solve these subproblems and combine them to obtain a global solution.



Solve the given problem via a series of locally optimal decisions and possibly revisit some decisions.

Question 10**1 / 1 pts**

The statements below concern both Prim's and Kruskal's algorithm for obtaining minimum spanning trees from an arbitrary graph $G = (V, E)$. Mark all statements below that are **true**.



At any given iteration of either algorithm, the partial solution obtained is necessarily a tree.

Correct!

The proof for Kruskal's algorithm seen in lecture applies the MST cycle property.

Correct!

At any given iteration of either algorithm, the partial solution obtained is necessarily a forest.



The proof for Prim's algorithm seen in lecture applies the MST cycle property.

Quiz Score: **10** out of 10