

Quiz - Backtracking and Greedy Algorithms

⚠ This is a preview of the draft version of the quiz

Instructions



This quiz will test your understanding of the material covered so far this week ([MLOs](#)).

This is an online quiz. There will be no time limit to the quiz. You can attempt the quiz twice and the best of the scores will be retained. This is open notes and open internet quiz but refrain from discussing with anybody during the exam.

Note that this test cannot be taken past the due date for any credit.



- Quiz Type** Graded Quiz
- Points** 10
- Assignment Group** Quizzes
- Shuffle Answers** Yes
- Time Limit** No Time Limit
- Multiple Attempts** Yes
- Score to Keep** Highest
- Attempts** 2
- View Responses** Always
- Show Correct Answers** After May 1 at 12am
- One Question at a Time** No

Due	For	Available from	Until
Apr 30	Everyone	-	Apr 30 at 11:59pm

Preview

Score for this attempt: **0** out of 10

Submitted May 3 at 8:55pm

This attempt took less than 1 minute.

Unanswered

Question 1

0 / 1 pts

What makes the solution for the 'Activity Selection Problem' that we implemented in the exploration, a greedy approach?

- ☐ It satisfies greedy property
- ☐ It is similar to Dynamic Programming algorithm
- ☐ It has optimal substructure

Correct Answer

- ☐ We make a best available choice in each iteration and we never look back



Unanswered

Question 2

0 / 3 pts

Pick the statements which are True.

Correct Answer

- ☐ Dynamic programming technique would always return an optimal solution

Correct Answer

- ☐ Greedy algorithms are efficient compared to dynamic programming algorithms
- ☐ Greedy algorithms would always return an optimal solution

Correct Answer☐

A greedy algorithm is hard to design sometimes as it is difficult to find the best greedy approach

Unanswered**Question 3****0 / 1 pts**

All possible greedy algorithms, at each step, choose what they know is going to lead to an optimal solution for the general problem.

Correct Answer☐ True☐ False**Unanswered****Question 4****0 / 1 pts**

Can 0/1 knapsack problem be solved using the Greedy algorithm technique to obtain an optimum solution to fill the knapsack?

0/1 knapsack problem (This is the problem that we saw in the previous modules) When have n items and their values given. We are provided with a knapsack of capacity X . We have only one copy of each item. We need to maximize the value of our knapsack with items that we pick.

☐ True**Correct Answer**☐ False

Greedy solution might not give us an optimal solution.

Unanswered

Question 5

0 / 3 pts

Fill in the below pseudocode for activity selection problem using the greedy approach. The function returns the count of the maximum number of activities that can be selected.

activitySelection(activities):

 sortBasedonEndTime(activities) # uses quick sort to sort the activities

 for activity in activities:

 if currentEndTime <= activity.startTime:

 [Select]

 [Select]

 return result

Time complexity for the pseudocode will be [Select]



Answer 1:

(You left this blank)

You Answered

Correct Answer

result = result + 1

Answer 2:

(You left this blank)

You Answered

Correct Answer

currentEndTime = activity.endTime

Answer 3:

(You left this blank)

You Answered

Correct Answer

$O(n^2)$

Quick sort has worst case of $O(n^2)$ time complexity

Unanswered

Question 6

0 / 1 pts

The asymptotic runtime of the solution for the combination sum problem that was discussed in the exploration is _____.

☐ N Factorial

☐ Exponential

☐ Logarithmic

☐ Linear

Correct Answer



Quiz Score: **0** out of 10