

CS3230 Semester 1 2025/2026  
Design and Analysis of Algorithms

**Tutorial 07**  
**Greedy Algorithms**  
**For Week 08**

Document is last modified on: October 2, 2025

## 1 Lecture Review: Greedy Algorithms

The basic idea of Greedy Algorithms (one of the important problem-solving paradigm that we learn in CS3230) is as follow: Cast the problem where we have to make a (single) **greedy choice** and are left with **one subproblem** to solve.

1. Prove<sup>1</sup> that there is always an **optimal solution to the original problem that makes the greedy choice**, so the greedy choice is safe. Most of the time, we use ‘exchange argument’ here.
2. Use **optimal substructure** to show that we can combine an optimal solution to the subproblem with the **greedy choice** to get an optimal solution to the original problem. Most of the time, we use ‘proof by contradiction’ (cut and paste argument) here, although ‘constructive proof’ can also be used.

In this tutorial, we will see a few more examples of problems that can be solved with greedy algorithms, on top of the few that we have seen in lecture.

## 2 Tutorial 07 Questions

Q1, Q2, and Q3). are related to the following **Burning CDs** problem:

Suppose Bob has a collection of music files that he wants to burn into CDs<sup>2</sup>. Each CD has a storage capacity of 100 MB. In addition, Bob does not want to store more than two<sup>3</sup> music files per CD. Note

---

<sup>1</sup>For theory course like CS3230, this is the most important part. In programming competitions involving greedy algorithms, some teams have ‘faith’ in their ‘greedy choice’, implement it, and hopes it passes all (secret) test cases.

<sup>2</sup>If you are not sure what is this, read <https://www.wikihow.com/Burn-a-CD>.

<sup>3</sup>This is indeed a weird constraint, but it is important for this problem to be amenable to greedy solution.

that each music file cannot be split and hence cannot be burned on more than 1 CDs. Given a set  $A$  of file sizes, each smaller than 100 MB, let  $MinCD(A)$  denote the minimum number of CDs required to fit the files described in  $A$ .

Q1). Which of the following correctly describes an optimal substructure property of the problem, assuming that **at least one pair of files fit into a CD**?

- (a) For any pair of files  $f_1, f_2$  in  $A$ ,  
 $MinCD(A) = 1 + MinCD(A \setminus \{f_1, f_2\})$
- (b) For any pair of files  $f_1, f_2$  in  $A$  that belong on a single CD in an optimal solution,  
 $MinCD(A) = 1 + MinCD(A \setminus \{f_1, f_2\})$
- (c) If  $f_1$  and  $f_2$  are the **largest** and **smallest** files in  $A$ ,  
 $MinCD(A) = 1 + MinCD(A \setminus \{f_1, f_2\})$

Q2). Assume that any optimal solution contains a pair that is burned onto a CD. Select all **True** statements about the problem.

- (a) The smallest file  $f$  must be included in a pair in some optimal solution.
- (b) The pair  $\{f_1, f_2\}$  where  $f_1$  is the smallest file and  $f_2$  is the largest file such that  $f_1$  and  $f_2$  fit onto one CD must be included in a pair in some optimal solution.
- (c) The pair  $\{f_1, f_2\}$  where  $f_1$  is the smallest file and  $f_2$  is the largest file must be included in a pair in some optimal solution.

Q3). Derive the greedy algorithm for obtaining the minimum number of CDs that Bob needs to burn his music files and apply it to this array of file sizes  $A = \{89, 74, 81, 12, 7, 91\}$ , what is the output of  $MinCD(A)$  for this test case?

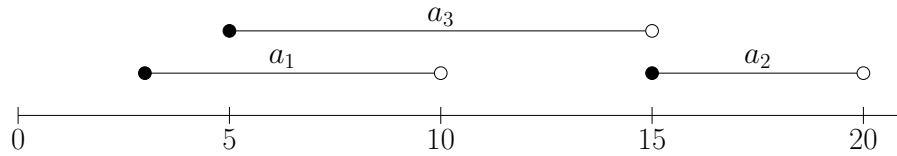
Q4, and Q5). are related to the following **Activity Selection** problem:

- You are given a set of activities  $S = \{a_1, a_2, \dots, a_n\}$ ,
- Each activity takes place during  $[s_i, f_i)$  (starting time – inclusive, finishing time – exclusive),
- Two activities  $a_i$  and  $a_j$  are **compatible** if their time intervals do not overlap, i.e.,  
 $s_i \geq f_j$  (activity  $a_i$  starts (right) after activity  $a_j$  finishes) or  
 $s_j \geq f_i$  (activity  $a_j$  starts (right) after activity  $a_i$  finishes).

Your task is to find the largest subset of mutually compatible activities.

Example:  $a_1 = [3, 10), a_2 = [15, 20), a_3 = [5, 15)$ , then:

- $\{a_1, a_2\}$  or  $\{a_2, a_3\}$  are compatible (and either are the optimal solutions).
- $\{a_1, a_3\}$  is not compatible.



Q4). Which of these greedy strategy (or strategies) work for the activity selection problem?

- (a) Choose the activity  $a$  that **starts last**, discard activities that conflict with  $a$ , and recurse.
- (b) Choose the activity  $a$  that **ends last**, discard activities that conflict with  $a$ , and recurse.
- (c) Choose the **shortest** activity  $a$ , discard activities that conflict with  $a$ , and recurse.

Q5). Derive the greedy algorithm for the Activity Selection Problem.

Optional Q6) - time permitting. Discuss the following LeetCode task during tutorial:

TA can choose to discuss in high-level only or show the partial/full code (in C++/Python/Java)

- Wednesday class: ipo
- Thursday class: non-overlapping-intervals
- Friday class: boats-to-save-people