1 List	Sorting &	Larching	Algos
			U

- 2) Find a partner + explain
 how I algo, works to each other
 (2 described in total)
- (3) New partner:
 Write pseudocode for one of
 the algorithms.
- (4) Next partners

 (betrace es explain (in a convincing way) either

 (a) why it terminates

 (b) numbers
- (5) Groups: (1) Algo. For finding common elements of @2 sorted 2 ways
 (2) Doesa Pseudocode
 - 3) Does it terminate?
 - 9) What if not sorted?

 (3) While loop on pa 207-termination 130 Aug 2019

Sorting problems come in two flavors: (1.) use sorting to make subsequent steps in an algorithm simpler, and (2.) design a custom sorting routine. For the former, it's fine to use a library sort function, possibly with a custom comparator. For the latter, use a data structure like a BST, heap, or array indexed by values.

Certain problems become easier to understand, as well as solve, when the input is sorted. The most natural reason to sort is if the inputs have a **natural ordering**, and sorting can be used as a preprocessing step to **speed up searching**.

For **specialized input**, e.g., a very small range of values, or a small number of values, it's possible to sort in O(n) time rather than $O(n \log n)$ time.

It's often the case that sorting can be implemented in **less space** than required by a brute-force approach.

Sometimes it is not obvious what to sort on, e.g., should a collection of intervals be sorted on starting points or endpoints? (Problem 13.5 on Page 210)

Table 13.1: Top Tips for Sorting

13.1 Compute the intersection of two sorted arrays

A natural implementation for a search engine is to retrieve documents that match the set of words in a query by maintaining an inverted index. Each page is assigned an integer identifier, its *document-ID*. An inverted index is a mapping that takes a word w and returns a sorted array of page-ids which contain w—the sort order could be, for example, the page rank in descending order. When a query contains multiple words, the search engine finds the sorted array for each word and then computes the intersection of these arrays—these are the pages containing all the words in the query. The most computationally intensive step of doing this is finding the intersection of the sorted arrays.

Write a program which takes as input two sorted arrays, and returns a new array containing elements that are present in both of the input arrays. The input arrays may have duplicate entries, but the returned array should be free of duplicates. For example, the input is $\langle 2,3,3,5,5,6,7,7,8,12 \rangle$ and $\langle 5,5,6,8,8,9,10,10 \rangle$, your output should be $\langle 5,6,8 \rangle$.

Hint: Solve the problem if the input array lengths differ by orders of magnitude. What if they are approximately equal?

Solution: The brute-force algorithm is a "loop join", i.e., traversing through all the elements of one array and comparing them to the elements of the other array. Let m and n be the lengths of the two input arrays.

The brute-force algorithm has O(mn) time complexity.

es flag products of content deputh rate of these setting for soils ordered very step to be set in the set of content of the set in t

Contain problems to oppose a complemental as well as well as each opposite as a contact contact as a contact contact as a contact contact as a conta

For epoclarized forms, as as we would range of earners, or a small mamber or visite as more than the state of an Or Larvi Sing.

it's own therefore that correspond be a upbromised in fees apace than required by a browning apparator.

Sometimes, this act obvious what to socious by singular discillent of infinite stable societies and an existed as

Printed and year out of the Science

CARRA GETRO-COLL TO STORMERS SPEED SERVICE TO A PERSON OF THE PROPERTY OF THE

A chiral a calculation les a centh origina is to retrieve documents the sector master master of words in a query by manutationing and two raids codes. Under page is existent an integer identifier his consent.

13. An interested indirection mapping that takes a word want of maps and tool areast of page into writing containing and a constitution of the containing and a containing and a containing and and then computed the other integer into a containing all the containing and the containing all the containing and and the containing all the containing all the containing and the containing and the containing and the containing all the containing and the c

while a preprint which takes as input who coved an est, and reluced the enter contracted of the contract of th

atem. Salvette e program d'arringen apper lengthe deller de netter et meganische d'ent if den eine expressione Annet

solytom The brace time steeniler is a "map ton" to exercise through at up the armiter of the language of the two costs and a feet to the costs of the two costs armin and the two costs armin armiters.

A compared to the combination of restable to the compared to the combined of t

- foldering that is the property of the analysis of

CONTROL AND ACCORDING TO THE CONTROL OF THE CONTROL

THE AND STREET, STREET, STREET,

the property of the second contract of the second s

Since both the arrays are sorted, we can make some optimizations. First, we can iterate through the first array and use binary search in array to test if the element is present in the second array.

The time complexity is $O(m \log n)$, where m is the length of the array being iterated over. We can further improve our run time by choosing the shorter array for the outer loop since if n is much smaller than m, then $n \log(m)$ is much smaller than $m \log(n)$.

This is the best solution if one set is much smaller than the other. However, it is not the best when the array lengths are similar because we are not exploiting the fact that both arrays are sorted. We can achieve linear runtime by simultaneously advancing through the two input arrays in increasing order. At each iteration, if the array elements differ, the smaller one can be eliminated. If they are equal, we add that value to the intersection and advance both. (We handle duplicates by comparing the current element with the previous one.) For example, if the arrays are $A = \langle 2, 3, 3, 5, 7, 11 \rangle$ and $B = \langle 3, 3, 7, 15, 31 \rangle$, then we know by inspecting the first element of each that 2 cannot belong to the intersection, so we advance to the second element of A. Now we have a common element, A, which we add to the result, and then we advance in both arrays. Now we are at A in both arrays, but we know A has already been added to the result since the previous element in A is also A. We advance in both again without adding to the intersection. Comparing A to A, we can eliminate A and advance to the fourth element in A, which is A, and equal to the element that A is iterator holds, so it is added to the result. We then eliminate A and since no elements remain in A, we return A.

Since we spend O(1) time per input array element, the time complexity for the entire algorithm is O(m+n).

mporti i dia sel dispensi serri serri altricito de la mandatativa de la completa de la completa de la completa Se estre agrada de migro esta de la completa de la Organización de la completa del completa de la completa de la completa del completa de la completa del la completa del la completa de la completa de la completa de la completa de la completa del la comple

The second secon

Francis sarount agrees

The Fire completifies of extor no vehicle in a the Corpletion for some field wind instant veter. We say
to the European operate size who by changing the shorter city for the potentially size of it is exact
another than an item reports on the countries that instance of the

party resided a real informace. As the entire with the real of the real production of the r

the course of the contract of the second second

The section of more standard and the section of the

Miller of the Control of the Control

a control of the signs of the control

el responsa acomo acomo atendos presentes del generale en comunitado en esta (13 laboras en