# **Developing Algorithms**

### Strategies -

- realize your problem is another well-known problem in disguise
  - it is searching or sorting
  - there's a data structure for that
  - it is a graph problem
- develop a new algorithm
  - divide and conquer
  - series of steps iterative
  - series of choices greedy, backtracking, branch and bound, dynamic programming

# **Iterative Patterns**

CPSC 327: Data Structures and Algorithms . Spring 2025

Iterative algorithms can be characterized by the main focus of the loop -

- process input
  - if the input is a collection of things, go through them one at a time
- produce output
  - if the output is a collection of things, produce them one at a time
  - if the output can be built incrementally, add to it one bit at a time
- narrow the search space
  - repeatedly eliminate elements that aren't the one you are looking for
  - repeatedly eliminate solutions that aren't the one you want
  - not applicable if you can't eliminate things without looking at them directly

#### CPSC 327: Data Structures and Algorithms . Spring 2025

# How to Design Algorithms

• establish the problem

CPSC 327: Data Structures and Algorithms . Spring 2025

- identify avenues of attack
- · define the algorithm
- show termination and correctness
- determine efficiency

## Algorithmic Structures

Iterative algorithms proceed forward towards the solution one step at a time.

repetition through loops

CPSC 327: Data Structures and Algorithms • Spring 2025

How to Design (Iterative) Algorithms						
Identify avenue • Paradigm Consider	tes of attack. <i>s and patterns.</i> the iterative patterns defined in section 3.1.					
pattern	loop structure					
process	input for each input element, process that element and incorporate it into the solution so far					
produce	output repeatedly produce the next output element repeatedly produce the next piece of the solution					
narrow search s	the repeatedly eliminate some non-solutions pace					
CPSC 327: Data Struc	ures and Algorithms • Spring 2025					

In a group of people, it is to be expected that some of them may not want to work with each other. Assuming that each person has at most d other people that they don't want to work with, divide the people into d+1 groups so that everyone is in exactly one group and no one is in a group with someone they don't want to work with.         Main steps. <u>petrem boop structure to work with.</u> <u>process input for each input element, process that element and incorporate it into the solution so far             incorporate the next input lement to obtain a solution for one are element.             for each person, put them into a group 2 respondents             do %             for each person, put them into a group not containing someone they             for each person, put them into a group as long as there isn't anyone in the             group that they don't want to work with             for each person, put them into a group as long as there isn't anyone in the             group that they don't want to work with             <b>Exit condition.</b>             The loop ends when             all of the input items have been processed             when all of the ouput items have been processed             when everyperson has been added to a group             3 respondents             do %             when everyperson has been added to a group             3 respondents             do %             when everyperson has been added to a group             3 respondents             do %             when everyperson has been added to a group             3 respondents             do %             when everyperson has been added to the current group             when all of the groups are full             when everyperson has been added to a group             3 respondents             do %             when everyperson has been added to a group             3 respondents             do %             when everyperson has been formed       </u>									
want to work with each other. Assuming that each person has at most d other people that they don't want to work with, divide the people into d+1 groups so that everyone is in exactly one group and no one is in a group with someone they don't want to work with.         Main steps.	In a group of people, it is to b								
most d other people that they don't want to work with, divide the people into d+1 groups so that everyone is in exactly one group and no one is in a group with someone they don't want to work with. <ul> <li>Pattern loop structure</li> <li>exit condition</li> <li>pattern loop structure</li> <li>exit condition so far</li> <li>incorporate the next input item into the solution to obtain a solution for one</li> <li>a respondents</li> <li>d0 %</li> <li>for each element, process it</li> <li>a respondents</li> <li>d0 %</li> <li>for each element, process it</li> <li>a respondents</li> <li>d0 %</li> <li>for each person, put them into the first group not containing someone they don't want to work with</li> <li>produce the next uput item</li> <li>for each person, put them into the group as there isn't anyone in the group that they don't want to work with</li> <li>Exit condition. The loop ends when</li> <li>all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the uput items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li>mean all of the input items have been processed</li> <li>a respondents</li> <li>d0 %</li> <li< td=""><td>want to work with each other</td><td></td><td></td><td></td><td></td><td></td></li<></ul>	want to work with each other								
people into d+1 groups so that everyone is in exactly one group and no one is in a group with someone they don't want to work with.       Image: Construction of the second	most d other people that they	don't want to	work with, divid	le the					
no one is in a group with someone they don't want to work with.         Main steps.       pattern       loop structure       exit condition         Main steps.       and oppart to the input element, process that element and incorporate it into the solution so far       when all of the input elements have been processed         incorporate the next input item into the solution to obtain a solution for one or element       3 respondents       60 %         for each element, process it       3 respondents       60 %         for each person, put them into the first group not containing someone they don't want to work with       5 respondents       100 %         produce the next output item for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with       0 %       3 respondents       60 %         Exit condition. The loop ends when       all of the input items have been produced       1 respondent       20 %       40 %         when every person has been added to a group       3 respondents       60 %       40 %	people into d+1 groups so that	at everyone is i	n exactly one gr	oup and					
Image: constraint of the input lemma input elements and incorporate it into the solution to obtain a solution for one element and incorporate it into the solution so far         Image: constraint of the input elements have been processed           Incorporate the next input lem into the solution to obtain a solution for one element         3 respondents         60 %           for each element, process it         3 respondents         60 %           for each element, process it         3 respondents         60 %           for each element, process it         3 respondents         60 %           for each person, put them into a group         2 respondents         40 %           for each person, put them into a group         2 respondents         60 %           for each person, put them into a group as long as there isn't anyone in the group hat they don't want to work with         3 respondents         60 %           Exit condition. The loop ends when         all of the input items have been produced         1 respondent         20 %           when every nerson has been added to a group         3 respondents         60 %         Image: secondents         100 %           when every one has been assigned to a legal group         5 respondents         00 %         Image: secondents         100 %           when one else can be added to the current group         0 %         Image: secondents         100 %         Image: secondents <td< td=""><td>no one is in a group with som</td><td>eone they don'</td><td>t want to work w</td><td>with.</td><td></td><td></td><td></td><td></td><td></td></td<>	no one is in a group with som	eone they don'	t want to work w	with.					
Main steps.       process input       for each input element, process that element, process that element and incorporate it into the solution so far solution so far solution so far and incorporate it into the solution to obtain a solution for one area helment, process it 3 respondents       60 %         for each element, process it       3 respondents       60 %         for each person, put them into a group       2 respondents       40 %         for each person, put them into the first group not containing someone they and two with       5 respondents       60 %         produce the next output item for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with       0 %       60 %         Exit condition. The loop ends when       all of the input items have been produced       1 respondents       60 %         all of the input items have been produced       1 respondents       60 %       60 %         when all of the ouput items have been produced       1 respondents       60 %         when all of the group as added to a group       3 respondents       60 %         when all of the group has been added to a group       3 respondents       60 %         when all of the group has been added to the current group       0 %       60 %         when all of the groups are full       0 %       60 %       60 %		pattern	loop structure		e:	xit conditi	ion		
incorporate the next input item into the solution to obtain a solution for one a respondents       3 respondents       60 %         for each element, process it       3 respondents       40 %         for each person, put them into the first group not containing someone they don't want to work with       5 respondents       100 %         produce the next output item       0 %       3 respondents       60 %         group hash bey don't want to work with       5 respondents       60 %         Exit condition. The loop ends when       3 respondents       60 %         all of the input items have been produced       1 respondent       20 %         when every person has been added to a group       3 respondents       60 %         when one one ise can be added to the current group       0 %       0 %         when one one ise can be added to the current group       0 %       0 %         when one of the groups are full       0 %       0 %	Main steps.	process input	ent, process t orate it into	that when all of the input elements have been the processed			e been		
for each element, process it       3 respondents       60 %         for each person, put them into a group       2 respondents       40 %         for each person, put them into the first group not containing someone they       5 respondents       100 %         produce the next output item       0 %       3 respondents       60 %         group that they don't want to work with       0 %       60 %       60 %         Exit condition. The loop ends when       all of the input items have been processed       3 respondents       60 %         all of the loupt items have been produced       1 respondents       60 %         when every person has been aadded to a legal group       3 respondents       60 %         when one else can be added to a legal group       5 respondents       100 %         when one else can be added to a legal group       5 respondents       00 %         when one else can be added to the current group       0 %       0 %         when all of the groups are full       0 %       0 %         when all of the groups have been formed       1 respondents       20 %	incorporate the next input item into more element	3 responder	nts (	50 %					
for each person, put them into the first group not containing someone they don't want to work with       2 respondents       40 %         produce the next output item for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with       0 %       3 respondents       60 %         Exit condition. The loop ends when       all of the input items have been processed       3 respondents       60 %         all of the input items have been produced       1 respondent       20 %         when every person has been added to a legal group       5 respondents       60 %         when one else can be added to a legal group       5 respondents       100 %         when no one else can be added to the current group       0 %       0 %         when all of the groups have been formed       1 respondents       20 %	for each element, process it			3 responder	nts e	50 %			
for each person, put them into the first group not containing someone they don't want to work with       5 respondents       100 %         produce the next output item for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with       0 %       3 respondents       60 %         Exit condition. The loop ends when       all of the input items have been produced       1 respondent       20 %         when all of the ouput items have been produced       1 respondents       60 %       60 %         when every person has been added to a group       3 respondents       60 %       60 %         when every no has been added to a group       3 respondents       60 %       60 %         when on one else can be added to the current group       0 %       0 %       60 %         when of the groups are full       0 %       0 %       60 %       60 %	for each person, put them into a gro	up		2 responder	nts 4	40 %			
produce the next output item     0 %       for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with     3 respondents     60 %       Exit condition. The loop ends when        all of the input items have been processed     3 respondents     60 %       when all of the ouput items have been produced     1 respondent     20 %       when every person has been added to a legal group     5 respondents     60 %       when no one else can be added to the current group     0 %       when one every propus ar full     0 %       when all of the groups are full     0 %	for each person, put them into the first group not containing someone they don't want to work with				nts 1	00 %		~	
for each group, add people to that group as long as there isn't anyone in the group that they don't want to work with       3 respondents       60 %         Exit condition. The loop ends when       all of the input items have been processed       3 respondents       60 %         all of the input items have been processed       3 respondents       60 %         when all of the ouput items have been produced       1 respondent       20 %         when every person has been added to a legal group       5 respondents       60 %         when no one else can be added to a legal group       5 respondents       100 %         when no one else can be added to the current group       0 %       0 %         when all of the groups are full       0 %       60 %	produce the next output item					0 %			
Exit condition. The loop ends when         all of the input items have been processed       3 respondents       60 %         when all of the ouput items have been produced       1 respondent       20 %         when every person has been assigned to a legal group       3 respondents       60 %         when every one has been assigned to a legal group       5 respondents       100 %         when one else can be added to the current group       0 %       0 %         when all of the groups are full       0 %       68	for each group, add people to that g group that they don't want to work w	3 responder	nts d	50 %					
all of the input items have been processed     3 respondents     60 %       when all of the ouput items have been produced     1 respondent     20 %       when every person has been added to a group     3 respondents     60 %       when everyone has been added to a legal group     5 respondents     100 %       when no one else can be added to the legal group     0 %     0 %       when no one else can be added to the urrent group     0 %     0 %       when all of the groups are full     0 %     63	Exit condition. The loop ends v	vhen							
when all of the ouput items have been produced     1 respondent     20 %       when every person has been added to a group     3 respondents     60 %       when everyone has been assigned to a legal group     5 respondents     100 %       when no one else can be added to the current group     0 %     0 %       when all of the groups are full     0 %     0 %       when did the groups have been formed     1 respondent     20 %	all of the input items have been proc	3 responden	nts 6	0 %					
when every person has been added to a group     3 respondents     60 %       when everyone has been assigned to a legal group     5 respondents     100 %       when no one else can be added to the current group     0 %     0 %       when all of the groups are full     0 %     68       when did the groups have been formed     1 respondent     20 %	when all of the ouput items have been produced				nt 2	0 %			
when everyone has been assigned to a legal group     5 respondents     100 %       when no one else can be added to the current group     0 %       when all of the groups are full     0 %       when d+1 groups have been formed     1 respondent     20 %	when every person has been added to a group				nts 6	0 %	~	·	
when no one else can be added to the current group     0 %       when all of the groups are full     0 %       when d+1 groups have been formed     1 respondent     20 %	when everyone has been assigned to a legal group			5 responden	nts 10	× 00			
when all of the groups are full 0 <sup>%</sup> when d+1 groups have been formed 1 respondent 20 <sup>%</sup>	when no one else can be added to the current group				(	» °			
when d+1 groups have been formed 1 respondent 20 %	when all of the groups are full				(	o %			_
	when d+1 groups have been formed	1 responden	nt 2	0 %			63		

# How to Design (Iterative) Algorithms

Define the algorithm. The core of an iterative algorithm is defining the loop.

Main steps.
 This is the core of the algorithm — the loop body. What's being repeated?

## • Exit condition. When does the loop end? pattern loop structure

East condition.							
When does the loop end?	pattern	loop structure	exit condition				
innen doeb ene loop endi	process input	for each input element, process that	when all of the input elements have bee				
		element and incorporate it into the	processed				
		solution so far					
	produce output	repeatedly produce the next output	when all of the output elements have				
		element	been produced				
		repeatedly produce the next piece of the	when the solution is complete				
		solution					
	narrow the	repeatedly eliminate some non-solutions	when the solution has found or there ar				
	search space		no solutions left				

- Setup. Whatever must happen before the loop begins.
- Wrapup. Whatever must happen to get the final answer after the loop ends.

Special cases.

- Make sure the algorithm works for all legal inputs identify the cases that need to be handled and address how that handling is incorporated into the previous steps (if not already accounted for).
- Algorithm.

61

Assemble the algorithm from the previous steps and state it.

There shouldn't be new elements here, instead bring together the main steps, exit condition, setup,

and wrapup along with any handling needed for special cases and state the whole algorithm.

62