LECTURE NO:4.1

19CSE100 Problem Solving and Algorithmic Thinking

Problem Definition and Comprehension

What is a Problem?

A doubtful or difficult matter requiring a solution

or

Something hard to understand or accomplish or deal with

- The Concise Oxford Dictionary (1995)

Reflections

How were you solving the problems?

What do you think is the reason for different solutions?

Reflections

Are all solutions equally good or one or few solutions better than others? How did you decide the goodness of the solution?

What do you think is/are needed for solving problems?

When do you think a problem is solvable (using computers)?

Making a cake

- Problem Statement : “ I need a cake for my grandmother’s birthday party.”

- Need to get information about making the cake:
  - What kind of cake does she like?
  - Does she have any allergies?
  - Does she have any health issues?
  - Any writing on the cake? Decorations? Candies

Solvable Problems

Objective answers through rational means; contain little/no ambiguity

Cake problem : Decision based on specific requirements

Cake making

Do we Need to get more information?

Is this relevant to finish the project ???

- Is the celebration with family members or you are sending invitation outside?
- Where is the party - indoors or outdoors?
- What is the date/time for the party?

Searching for a solution

So now shall we see about more data to help us come up with a more specific potential solution

Data collection

- Taste & Style?
- Allergies?
- Health issues?
- Presentation?
- Party context:
- Budget

- She likes chocolate and fruit.
- Nuts
- None
- A simple “Happy Birthday” with simple decorations, flowers and a base. No candles.
- Indoor - 15 members
- Rs. 8000/-

Iteration

- “I need a cake for my grandmother’s birthday party”

becomes

- I need a birthday cake for my grandmother’s birthday party. The cake should include chocolate and fruit, and have a simple frosting, and it should not include nuts. The cake should include some flowers and say Happy Birthday !”

Solvable Problems

Well structured; contains sufficient information for a solution to be found

Problem Understanding

Dissect problem into components (problems and sub-problems)

Problem Understanding

Remove ambiguity and extraneous information

Problem Understanding

Determine knowns and unknowns
Be aware of your assumptions

DECOMPOSITION

Smaller sub-problems

I NEED A CAKE FOR MY GRANDMOTHER'S BIRTHDAY PARTY

- Cake should include chocolate and fruit
- Presentation: Some flowers and piping, "Happy Birthday" and no candles
- The frosting should be simple without nuts
- The cake should be transported to her retirement home on May 4th for a 4pm indoor party
- Build a cake base decorated with some photos
- The cake should feed 15 people

Deadline: May 3
Budget: Rs. 8000/-

DECOMPOSITION

Smaller sub-problems

I NEED A CAKE FOR MY GRANDMOTHER'S BIRTHDAY PARTY

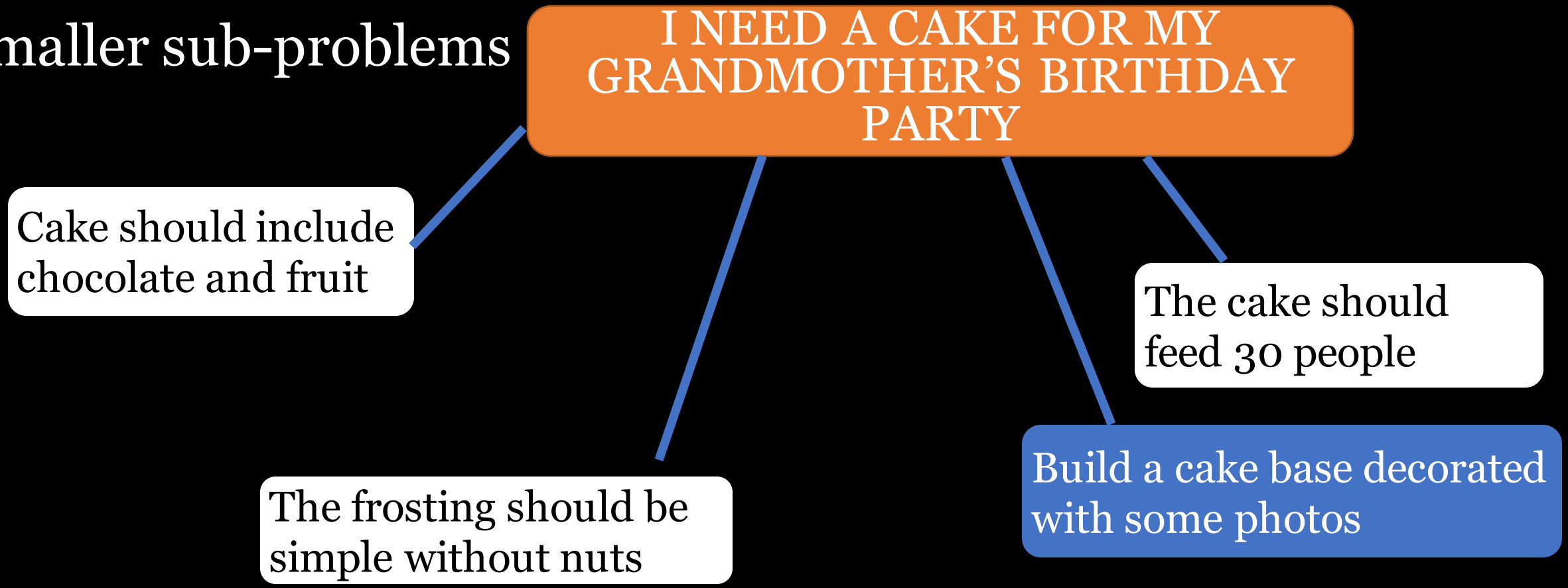
- Cake should include chocolate and fruit
- The frosting should be simple without nuts
- The cake should feed 30 people

What are some algorithms for the cake and frosting? Any patterns we can use?

Deadline: May 3
Budget: Rs. 8000/-

DECOMPOSITION / ABSTRACTION

Smaller sub-problems



What are some algorithms for building the cake base? Is this now too much? Should we remove the fancy base and focus on the cake?

Deadline: May 3
Budget: Rs. 8000/-

DECOMPOSITION

Smaller sub-problems

I NEED A CAKE FOR MY GRANDMOTHER'S BIRTHDAY PARTY

- Cake should include chocolate and fruit
- Presentation: Some flowers and piping, "Happy Birthday" and no candles
- The frosting should be simple without nuts
- The cake should be transported to her retirement home on May 4th for a 4pm indoor party
- Build a cake base decorated with some photos
- The cake should feed 30 people

Deadline: May 3
Budget: Rs. 8000/-

What is the algorithm for transporting the cake?

END OF LECTURE 4.1

LECTURE 4.2

Moving to a Solution

How to Solve the problem?

- Identifying pattern
- Have we solved similar kind of problem?
- What are different ways we can find solution?

Moving to a solution

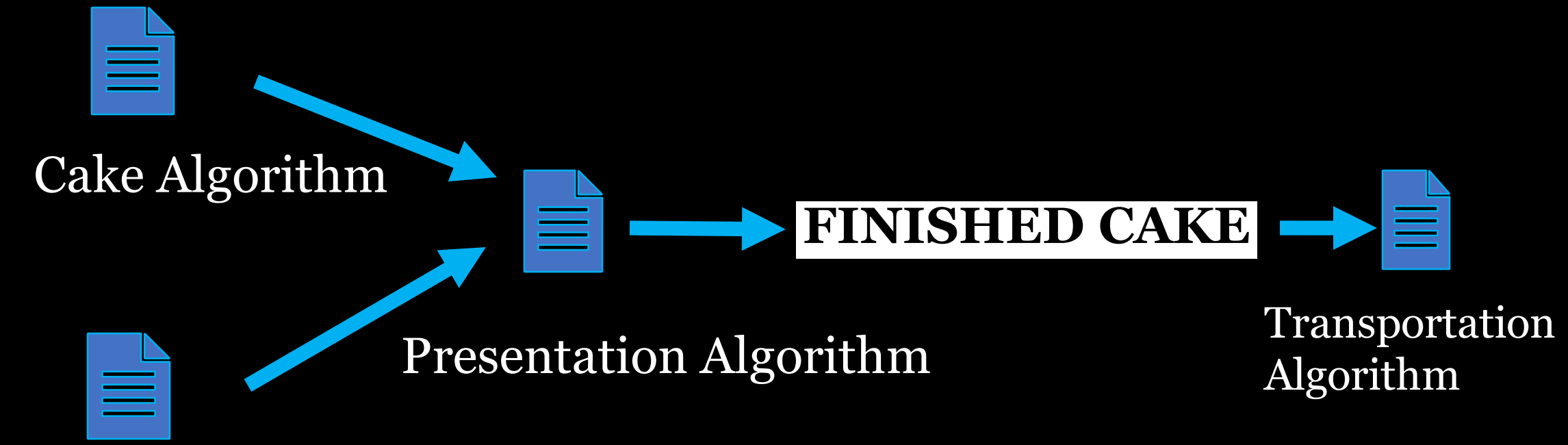
"I need a cake for my grandmother's birthday party."

Budget: Rs. 8000/-

Now, you could think about turning these "algorithms" into a more specific set of instructions((or programs) for some computational agent to enact.

Moving to solution

"I need a cake for my grandmother's birthday party."



Corner stones

- Problem Identification: Can a computer help solve my problem? Is this problem suited for a computer?
- Decomposition: Do I need to break down my problem into smaller pieces to help solve it? How do the situations to the smaller pieces fit together to solve the main problem?
- Pattern Recognition: Have I seen problem - solving approaches in the past that can help me solve one of the problems or sub - problems that I have there?
- Abstraction: Are the aspects of your problem that are not necessarily needed now to solve your problem?

Problem identification

- Topic: What is the big problem you are trying to solve? Can you state this in a problem statement? Is the problem statement too broad or vague?
- Data: What information or data do you have to contribute to a problem-solving approach? What other information or data might you need?
- Feasibility: Can you solve the problem given what you know ?Is this problem solvable with a computer?

Problem identification

- Have we identified a problem that is “solvable” with a computer? Believe it or not, some problems can’t be necessarily solved with computers!
- How can we know when a problem is a good candidate for using a computer to generate a solution?
- How might we incorporate different kinds to solve this type of problem?

Abstraction

- Can we identify non - essential aspects of our problem that we can ignore?
- This involves filtering out, or ignoring, aspects of the problem or any patterns you might have identified in order to concentrate on the necessary aspects of the problem or potential solution.
- Think about…
  - Asking whether some parts of the problem less relevant to a successful outcome? Are any extraneous?
  - Identifying instances when considering unnecessary or too many components of a problem can lead to situations where you are trying to do too much at the risk of an effective problem solution.

Decomposition

- You have a problem statement… can we break our big problem down into smaller sub - problems that are more straightforward to solve?
- If the problem statement is too big, it is difficult to see how you can go about developing a conceptual solution.
- If you can decompose your big problem into smaller and smaller sub - problems, you can start to think about how you can solve those smaller, specific sub-problems.
- Solving the sub-problems can then help you arrive at a solution for your original problem statement.

Pattern recognition

- Can we see familiar patterns or characteristics that can lead us to potential solutions?
- Finding the similarities or patterns amount the smaller, decomposed problems that can help us solve these problems more efficiently.
- Think about…
  - Have you seen solutions to similar problems previously that we can use here? Can you think of similar problems that have been solved before that could help you address these problems?
  - How is this problem the same or different from other problems you have identified or addressed?

Concluding remarks

- So after iterating through the process of framing all the four mentioned stepping stones of computational thinking, that's the evolution of "things to think about".

- And now we are ready to develop a plan to solve our problem in a way that we can then start writing the computer program.

Reference

- Michigan State University Materials