


CSE 101
Slide Set 7

Doç. Dr. Mehmet Göktürk
Department of Computer Engineering

www.gtu.edu.tr

1




Algorithms

- The Concept of an Algorithm
- Algorithm Representation
- Algorithm Discovery
- Iterative Structures
- Recursive Structures
- Efficiency and Correctness

www.gtu.edu.tr

CSE 101 Slide Set 7

2




Definition of Algorithm

An algorithm is an **ordered** set of **unambiguous, executable** steps that defines a **terminating** process.

www.gtu.edu.tr

CSE 101 Slide Set 7

3



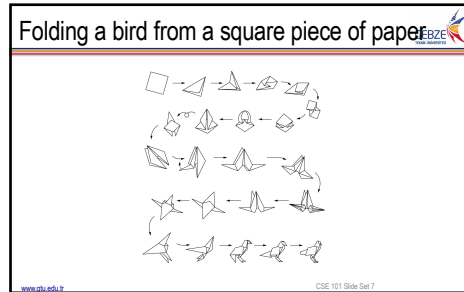
Algorithm Representation

- Requires well-defined primitives
- A collection of primitives constitutes a programming language.

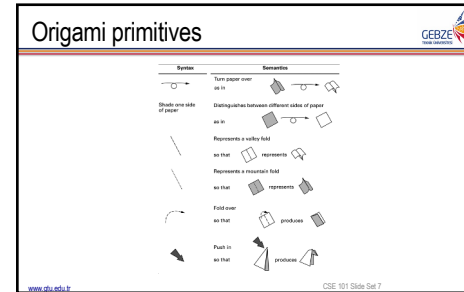
www.gtu.edu.tr

CSE 101 Slide Set 7

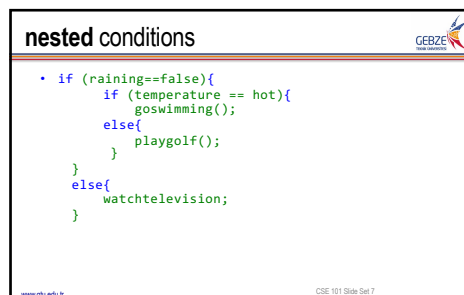
4



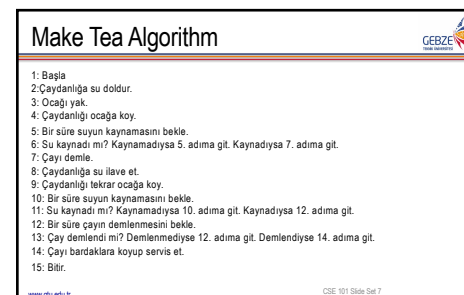
5



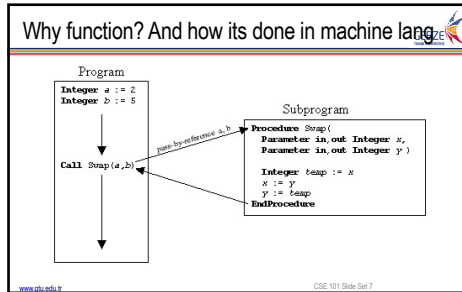
6



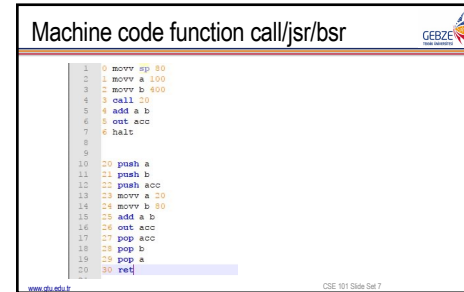
7



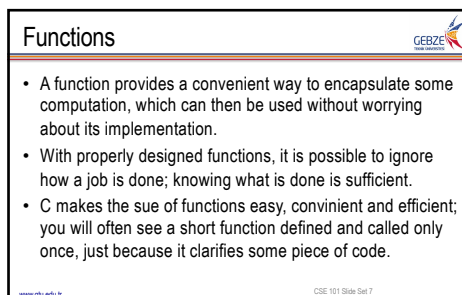
8



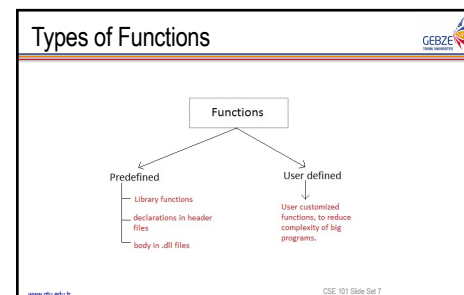
13



14

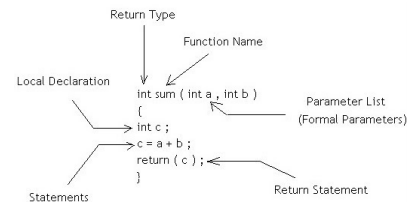


15



16

Details of Functions



17

Function Examples and Q&A

18

Pseudocode Primitives / Functions

- Using parameters

```
int Sort(List);
.
.
```
- Executing Sort on different lists

```
Sort(the membership list);
Sort(the wedding guest list);
```

Polya's Problem Solving Steps



- 1. Understand the problem.
- 2. Devise a plan for solving the problem.
- 3. Carry out the plan.
- 4. Evaluate the solution for accuracy and its potential as a tool for solving other problems.

www.dtu.edu.tr

CSE 101 Slide Set 7

21

Polya's Steps in the Context of Program Development



- 1. Understand the problem.
- 2. Get an idea of how an algorithmic function might solve the problem.
- 3. Formulate the algorithm and represent it as a program.
- 4. Evaluate the solution for accuracy and its potential as a tool for solving other problems.

www.dtu.edu.tr

CSE 101 Slide Set 7

22

Getting a Foot in the Door



- Try working the problem backwards
- Solve an easier related problem
 - Relax some of the problem constraints
 - Solve pieces of the problem first (bottom up methodology)
- Stepwise refinement: Divide the problem into smaller problems (top-down methodology)

www.dtu.edu.tr

CSE 101 Slide Set 7

23

Ages of Children Problem



- Person A is charged with the task of determining the ages of B's three children.
 - B tells A that the product of the children's ages is 36.
 - A replies that another clue is required.
 - B tells A the sum of the children's ages.
 - A replies that another clue is needed.
 - B tells A that the oldest child plays the piano.
 - A tells B the ages of the three children.
- How old are the three children?

www.dtu.edu.tr

CSE 101 Slide Set 7

24

Figure 5.5

a. Triples whose product is 36		b. Sums of triples from part (a)	
(1,1,36)	(1,6,6)	$1 + 1 + 36 = 38$	$1 + 6 + 6 = 13$
(1,2,18)	(2,2,9)	$1 + 2 + 18 = 21$	$2 + 2 + 9 = 13$
(1,3,12)	(2,3,6)	$1 + 3 + 12 = 16$	$2 + 3 + 6 = 11$
(1,4,9)	(3,3,4)	$1 + 4 + 9 = 14$	$3 + 3 + 4 = 10$

25

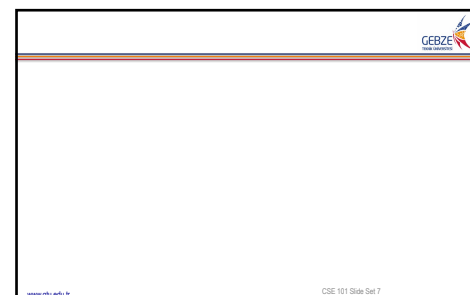
The sequential search algorithm in pseudocode

```

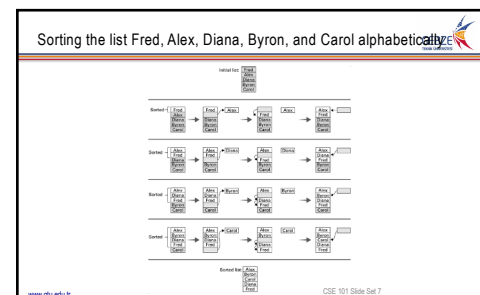
def Search (List, TargetValue):
    if (List is empty):
        Declare search a failure
    else:
        Select the first entry in List to be TestEntry
        while (TargetValue > TestEntry and entries remain):
            Select the next entry in List as TestEntry
        if (TargetValue == TestEntry):
            Declare search a success
        else:
            Declare search a failure

```

26



27



28

The insertion sort algorithm expressed in pseudocode

```

def Sort(List):
    N = 2
    while (N <= length of List):
        Pivot = Nth entry in List
        Remove Nth entry leaving a hole in List
        while (there is an Entry above the
            hole and Entry > Pivot):
            Move Entry down into the hole leaving
            a hole in the list above the Entry
        Move Pivot into the hole
        N = N + 1

```

29

Recursion

- The execution of a procedure leads to another execution of the procedure.
- Multiple activations of the procedure are formed, all but one of which are waiting for other activations to complete.

30

Applying our strategy to search a list for the entry John

Original list	First sublist	Second sublist
Alice Bob Carol David Elaine Fred George Harry Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly

31

A first draft of the binary search technique

```

if (List is empty):
    Report that the search failed
else:
    TestEntry = middle entry in the List
    if (TargetValue == TestEntry):
        Report that the search succeeded
    if (TargetValue < TestEntry):
        Search the portion of List preceding TestEntry for
        TargetValue, and report the result of that search
    if (TargetValue > TestEntry):
        Search the portion of List following TestEntry for
        TargetValue, and report the result of that search

```

32

The binary search algorithm in pseudocode

```
def Search(List, TargetValue):
    if (List is empty):
        Report that the search failed
    else:
        TestEntry = middle entry in the List
        if (TargetValue == TestEntry):
            Report that the search succeeded
        if (TargetValue < TestEntry):
            Sublist = portion of List preceding TestEntry
            Search(Sublist, TargetValue)
        if (TargetValue > TestEntry):
            Sublist = portion of List following TestEntry
            Search(Sublist, TargetValue)
```

www.dtu.edu.tr CSE 101 Slide Set 7

33

Search Pseudocode

We are here.

```
def Search(List, TargetValue):
    if (List is empty):
        Report that the search failed
    else:
        TestEntry = middle entry in List
        if (TargetValue == TestEntry):
            Report that the search succeeded
        if (TargetValue < TestEntry):
            Sublist = portion of List preceding TestEntry
            Search(Sublist, TargetValue)
        if (TargetValue > TestEntry):
            Sublist = portion of List following TestEntry
            Search(Sublist, TargetValue)
```

www.dtu.edu.tr CSE 101 Slide Set 7

34

Continued..

We are here.

```
def Search(List, TargetValue):
    if (List is empty):
        Report that the search failed
    else:
        TestEntry = middle entry in List
        if (TargetValue == TestEntry):
            Report that the search succeeded
        if (TargetValue < TestEntry):
            Sublist = portion of List preceding TestEntry
            Search(Sublist, TargetValue)
        if (TargetValue > TestEntry):
            Sublist = portion of List following TestEntry
            Search(Sublist, TargetValue)
```

www.dtu.edu.tr CSE 101 Slide Set 7

35

Continued ...

We are here.

```
def Search(List, TargetValue):
    if (List is empty):
        Report that the search failed
    else:
        TestEntry = middle entry in List
        if (TargetValue == TestEntry):
            Report that the search succeeded
        if (TargetValue < TestEntry):
            Sublist = portion of List preceding TestEntry
            Search(Sublist, TargetValue)
        if (TargetValue > TestEntry):
            Sublist = portion of List following TestEntry
            Search(Sublist, TargetValue)
```

www.dtu.edu.tr CSE 101 Slide Set 7

36

Algorithm Efficiency

- Measured as number of instructions executed
- Big theta notation: Used to represent efficiency classes
 - Example: Insertion sort is in $\Theta(n^2)$
- Best, worst, and average case analysis

www.dtu.edu.tr

CSE 101 Slide Set 7

37

Applying the insertion sort in a worst-case situation

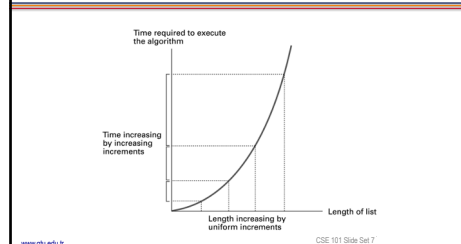
Comparisons made for each pivot					
Initial list	1st pivot	2nd pivot	3rd pivot	4th pivot	Sorted list
Elaine David Carol Barbara Alfred	1 Elaine David Carol Barbara Alfred	3 David 2 Elaine Carol Barbara Alfred	6 Carol David 5 Elaine 4 Barbara Alfred	10 Barbara Carol 9 David 8 Elaine 7 Alfred	Alfred Barbara Carol David Elaine

www.dtu.edu.tr

CSE 101 Slide Set 7

38

Graph of the worst-case analysis of the insertion sort algorithm

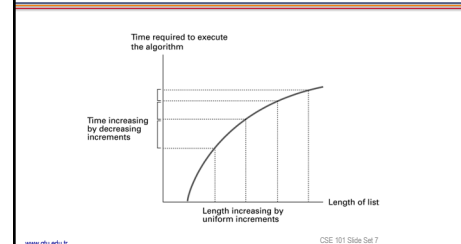


www.dtu.edu.tr

CSE 101 Slide Set 7

39

Graph of the worst-case analysis of the binary search algorithm



www.dtu.edu.tr

CSE 101 Slide Set 7

40

Software Verification

- Proof of correctness
 - Assertions
 - Preconditions
 - Loop invariants
- Testing

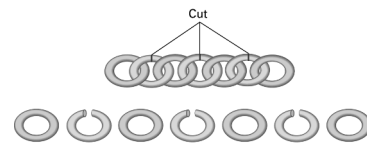
41

Chain Separating Problem

- A traveler has a gold chain of seven links.
- He must stay at an isolated hotel for seven nights.
- The rent each night consists of one link from the chain.
- What is the fewest number of links that must be cut so that the traveler can pay the hotel one link of the chain each morning without paying for lodging in advance?

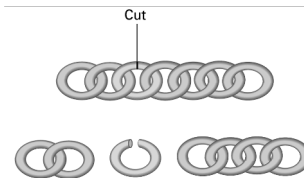
42

Separating the chain using only three cuts



43

Solving the problem with only one cut




44

Solving the problem with only one cut

- Night 1: Gives 1 link
- Night 2: Takes back single link and gives 2 links
- Night 3: Gives single link
- Night 4: Takes back all of the links and gives the 4 links
- Night 5: Gives the single link
- Night 6: Takes back the single link and gives the double
- Night 7: Gives the single link

www.gdu.edu.tr CSE 101 Slide Set 7

45




www.gdu.edu.tr CSE 101 Slide Set 7

46



www.gdu.edu.tr CSE 101 Slide Set 7

47



www.gdu.edu.tr CSE 101 Slide Set 7

48

