Programming Studio #10

ECE 190

Programming Studio #10

- MP4.1
  - Due Wednesday, April 7th at 5PM
  - Start early!
  - Read handout carefully
  - Cheating Policy is on the website

Recursion

- A function calls itself.
- Pros:
  - Better solution for some problems
    - Searching
  - Fewer lines of code and elegant logic
- Cons:
  - Not an easy programming concept
  - Memory overhead (stack frames)

How to Write a Recursion

- Composition
  - Base case: the termination condition
  - Recursive step: the intermediate step
- Implement a recursion is easy
  - Recall computing factorial, Fibonacci number generator, etc.
- Design a recursion algorithm is not easy
  - Identify recursive patterns

Example: Binary Search

- Problem: find a key `x` in a sorted array `sorted_array`
- Algorithm:
  - Recursive step
    - Find the middle number `m` of the array
    - If the key is greater than `m`, search the right half.
    - If the key is smaller than `m`, search the left half.
  - Base case
    - If the key equals `m`, return `m`'s position
    - If the array size is zero, return -1

Binary Search Code

```
1. int binary_search (int sorted_array[], int first, int last,
   int key)
2. {
3.   int mid=(first+last)/2;
4.   if (first == last)
5.     return -1;  // array size zero
6.
7.   if (key == sorted_array[mid])
8.     return mid;  // find the key

9.   if (key < sorted_array[mid]) // search the left half
10.    return binary_search(sorted_array, first, mid-1, key);
11.  else  // search the right half
12.    return binary_search(sorted_array, mid, last, key);
13.}
```

Exercise – Maze Walk

- Find if a one can exit a maze from a chosen position.
- Download the file `maze.c` from the website.
- Hints:
  - If a position (x,y) can exit the maze, the nearby position (x±1, y±1), if valid and accessible from (x,y), can also exit the maze.
  - Each move is a recursive step.
  - Remember visited places to avoid endless loop.