Midterm Summary

CSCI2100A Data Structures

April 11st, 2016

Written Midterm

- Participant: 122
- Make-up: 2
- Absent: 0

Summary of results

- Mean: 81.3
- Standard deviation: 13.7
- Highest: 100 (5 students)

- You can check your paper at SHB 1024 during office hours (don’t take it away).

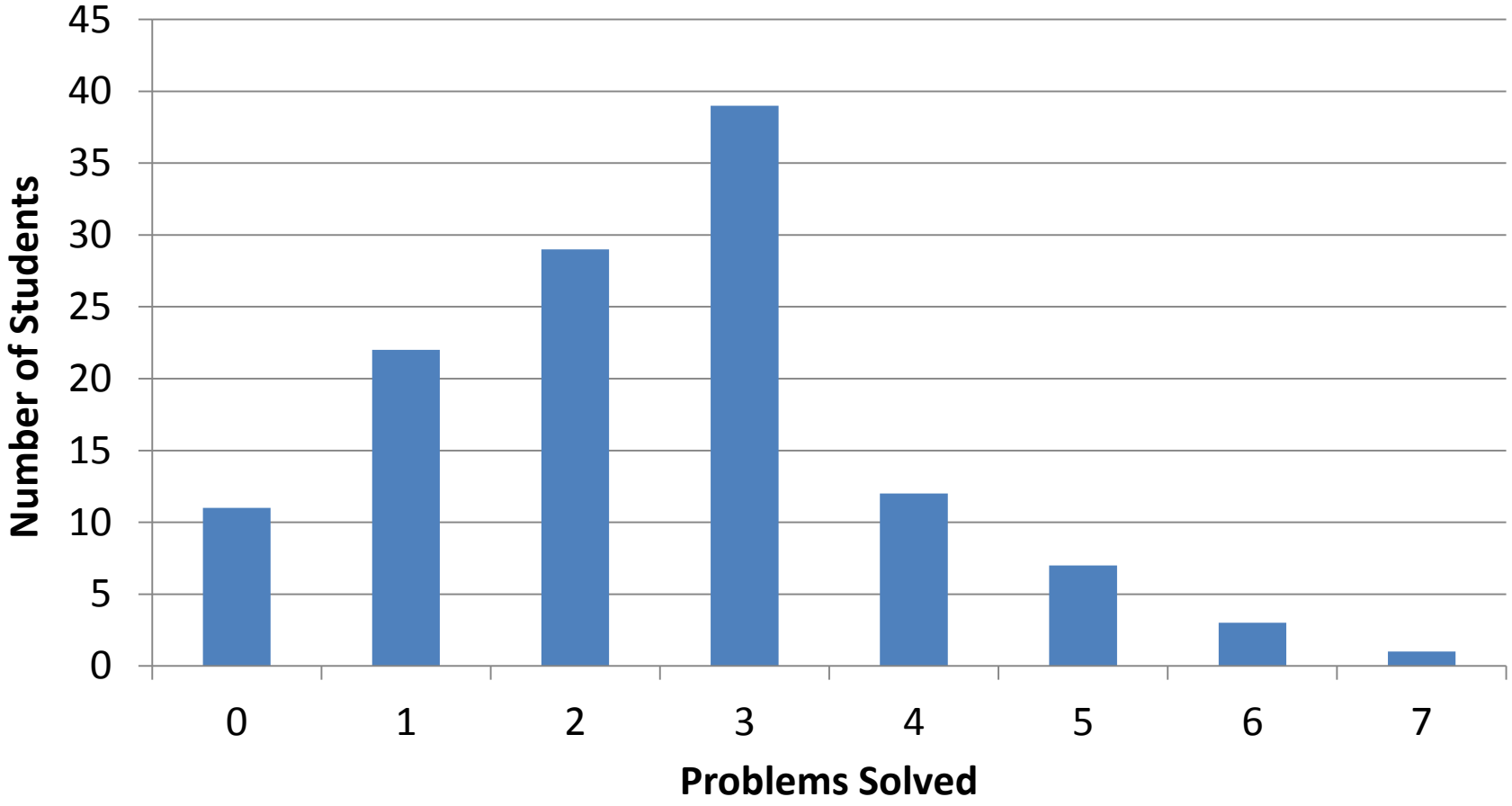
Programming Midterm

- Participant: 124
- Absent: 0
- Afternoon session: 7
- Evening session: 117
- Total attempt/solved: 808/305
- Average solved: 2.5

Summary of results

Problem Solved	Number of Students
7	1
6	3
5	7
4	12
3	39
2	29
1	22
0	11

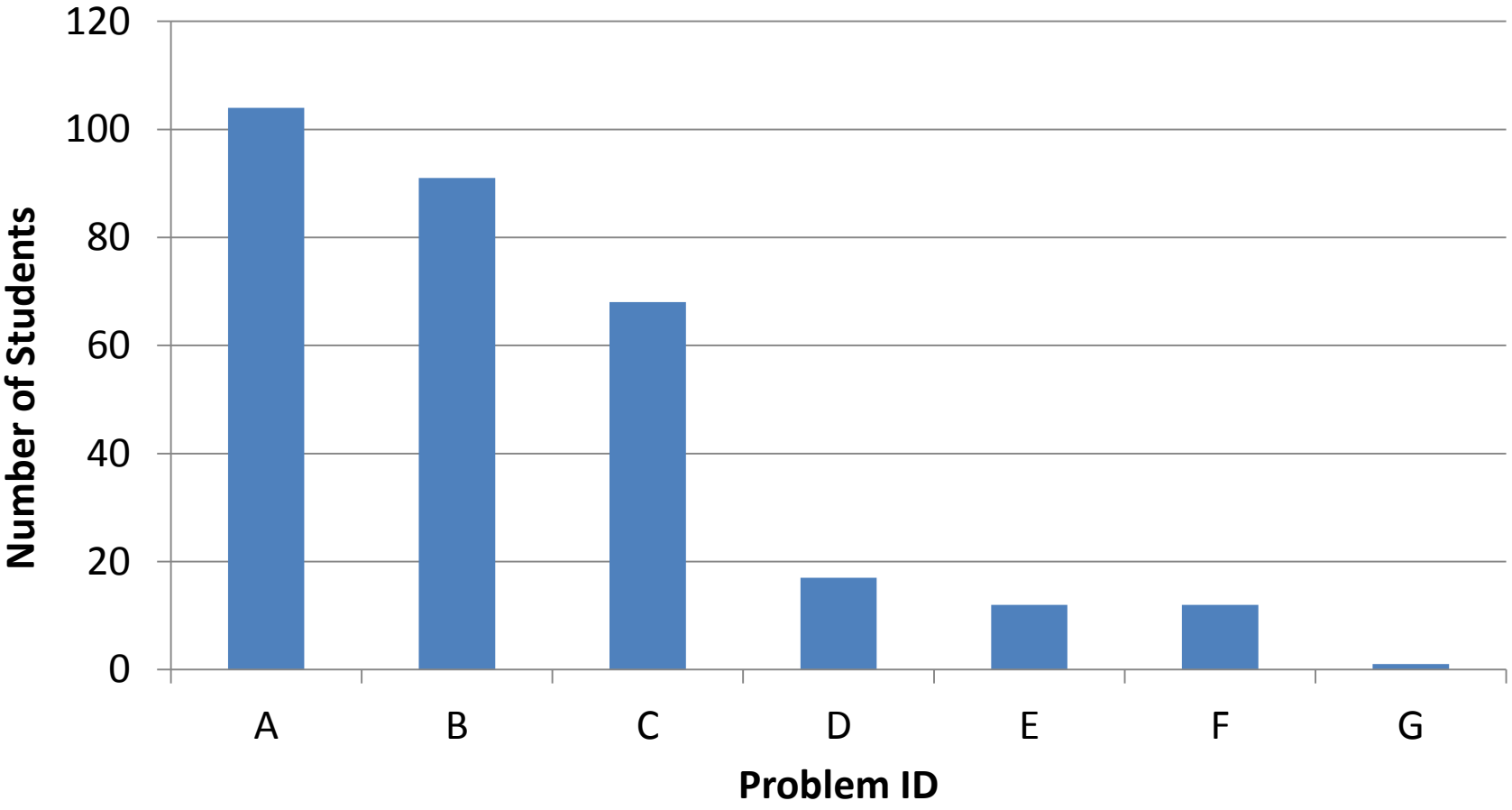
Summary of results (cont.)



Summary of results (cont.)

Problem ID	Number of Students
A	104
B	91
C	68
D	17
E	12
F	12
G	1

Summary of results (cont.)



A – Word Game

- Difficulty: Easy

- String manipulation
- One for-loop

B – Number Pyramid

- Difficulty: Easy

- Array
  - 2 one-dimension arrays are enough. Why?
- Nested for-loops

C – Valid Parentheses

- Difficulty: Easy

- Stack
  - push(x)
  - pop()
  - isEmpty()

D – Lone Survivor

- Difficulty: Medium

- $n \leq 1000$ => $O(n^3)$ algorithm is acceptable
  - Linked list (deletion)
  - Array
    - person[]: 1 to n
    - deleted[]: 0 or 1
- Josephus problem
  - Best time complexity $O(n^2)$ with DP

E – Colors of Balloons

- Difficulty: Medium

- String hashing: time complexity $O(n)$
- One example of hash function
  - Given string s
  - $f(s) = (\sum_i int(s[i]) * (i + 1)) \; mod \; P$

F – Binary Tree Level Order Traveral

- Difficulty: Medium

- Tree traversal: given pre-order and in-order, you need to reconstruct the tree structure
- Queue
  - insert root into queue, cur_pos = 0
  - insert left node and right node of queue[cur_pos] into queue (if any)
  - cur_pos += 1

G – Find Median from Data Stream

- Difficulty: Hard

- The key idea is to use two heaps to store the small half and big half
- Always keep small.size()=large.size() **or** small.size()=large.size()+1
- How to perform addNum(x) and findMedian()?