University of Colorado
Boulder

CSCI 3104 Algorithms

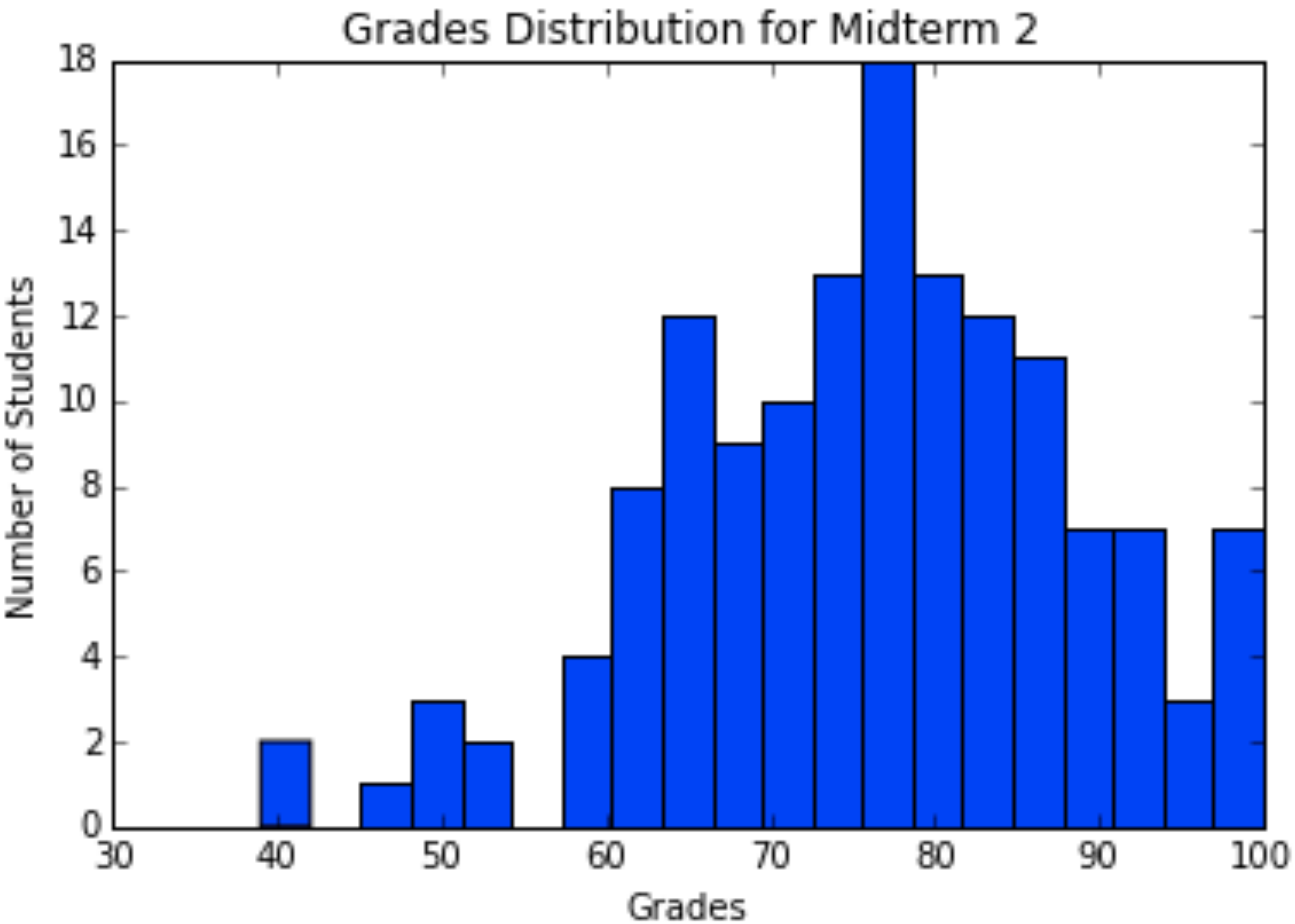
Fall 2015
Lecture 31 (Nov 9)

Announcements

- Homework 1-7, Midterm Exam 1 & 2
  - pick up graded homework
  - double check scores at moodle
- Homework 8 (last homework)
  - will be posted on Wednesday, Nov 11
  - due at 11am, Wednesday, Nov 18
- Final Exam
  - Th, Dec 17, 4:30pm -- 7:00pm, FLMG 155

Midterm Exam 2

- Q1: 4 x 5 = 20
- Q2: 15 + 15 + 10 = 40
- Q3: 2 x 15 = 30
- Q4: 10
- Median = 77.0
- Mean = 76.0
- Stdev = 12.2



Problem 1(a)

- Determine if the following statements are TRUE or FALSE. Briefly explain why.
  - (a) Given a graph G = (V, E) with positive edges, to find the shortest paths from starting vertex s to all vertices in the graph, Dijkstra's shortest-path algorithm requires O(|V|) delete-min() operations.
  - TRUE. Each vertex is removed exactly once via the delete-min() operation.

Problem 1(b)

- Determine if the following statements are TRUE or FALSE. Briefly explain why.
  - (b) The greedy algorithm for the knapsack problem can always find the optimal solution for both fractional knapsack and 0-1 knapsack.
  - FALSE. The greedy algorithm is optimal for fractional knapsack, but may not be optimal for 0-1 knapsack.

Problem 1(c)

- Determine if the following statements are TRUE or FALSE. Briefly explain why.
  - (c) Dynamic programming algorithms cannot guarantee optimal solution.
  - FALSE. Dynamic programming algorithms check all possible solutions, thus can guarantee optimal solution.

Problem 1(d)

- Determine if the following statements are TRUE or FALSE. Briefly explain why.
  - (d) A linear programming problem is expressed solely by its variables and objective function.
  - FALSE. Linear programming problems also need to specify the constraints.

Problem 2(a)

- Show the key steps of the following tasks:
  - (a) Huffman coding of 7 symbols with the following frequencies:
  - {A: 11, B: 26, C: 8, D: 15, E:13, F: 10, G: 17}
  - C:8 + F:10 => CF: 18   A:11 + E:13 => AE: 24
  - D:15 + G:17 => DG: 32
  - CF:18 + AE: 24 => ACEF: 42
  - B:26 + DG:32 => BDG: 58 + ACEF: 42 => 100

Problem 2(b)

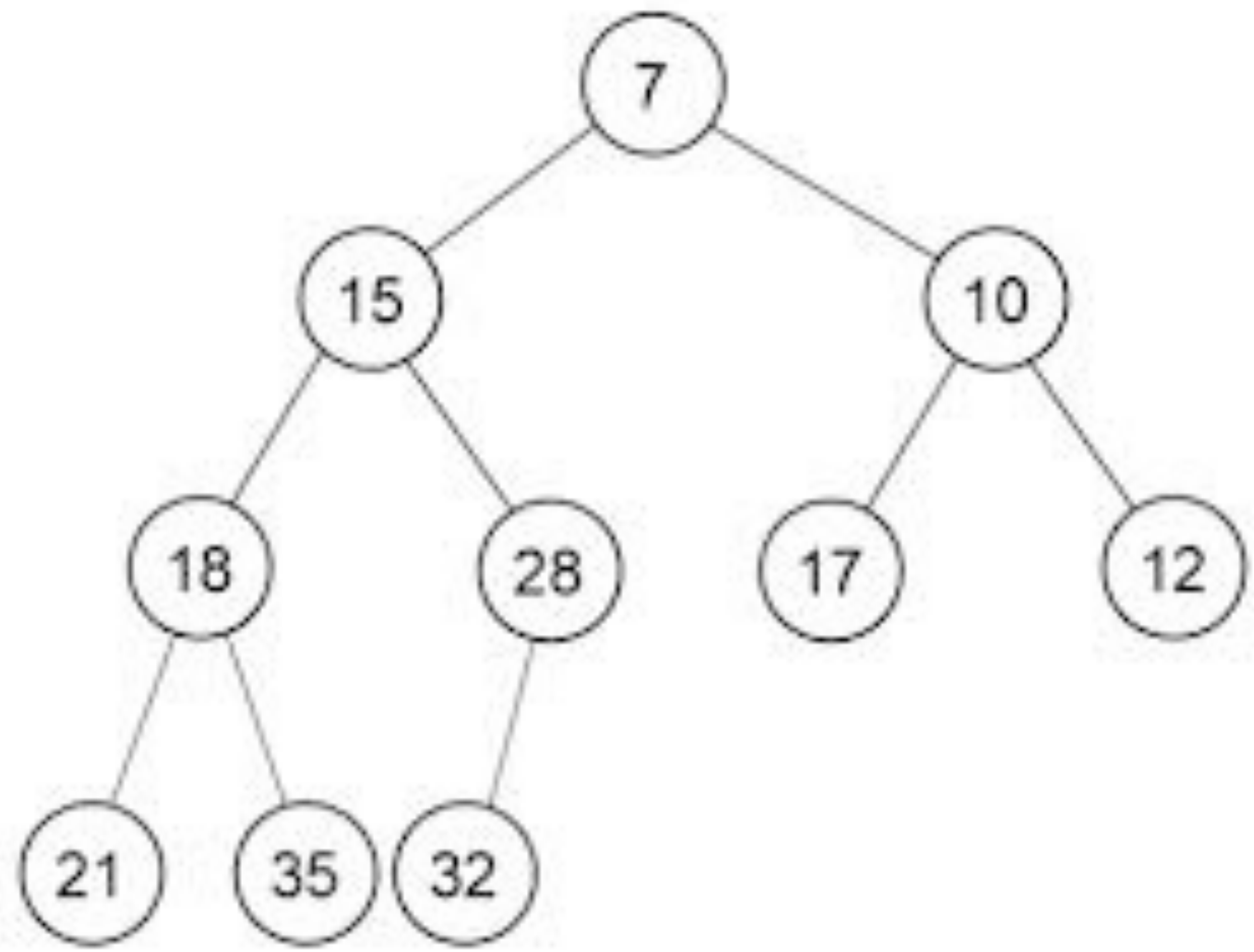
- (b) Formulate a linear programming problem for the diet problem: A nutritionist is planning a menu consisting of two main foods A and B. Each ounce of A contains 2 units of fat, 1 unit of carbohydrate, and 4 units of protein. Each ounce of B contains 3 units of fat, 3 units of carbohydrates, and 3 units of protein. The nutritionist wants the meal to provide at least 18 units of fat, at least 12 units of carbohydrates, and at least 24 units of protein. If an ounce of A costs 20 cents and an ounce of B costs 25 cents, how many ounces of each food should be served to minimize the cost of meal yet satisfy the nutritionist's requirements?

Problem 2(b)

- Linear Programming
  - variables, constraints, objective function
- $x_a, x_b >= 0$
- $2x_a + 3x_b >= 18$
- $1x_a + 3x_b >= 12$
- $4x_a + 3x_b >= 24$
- $\min 20x_a + 25x_b$

Problem 2(c)

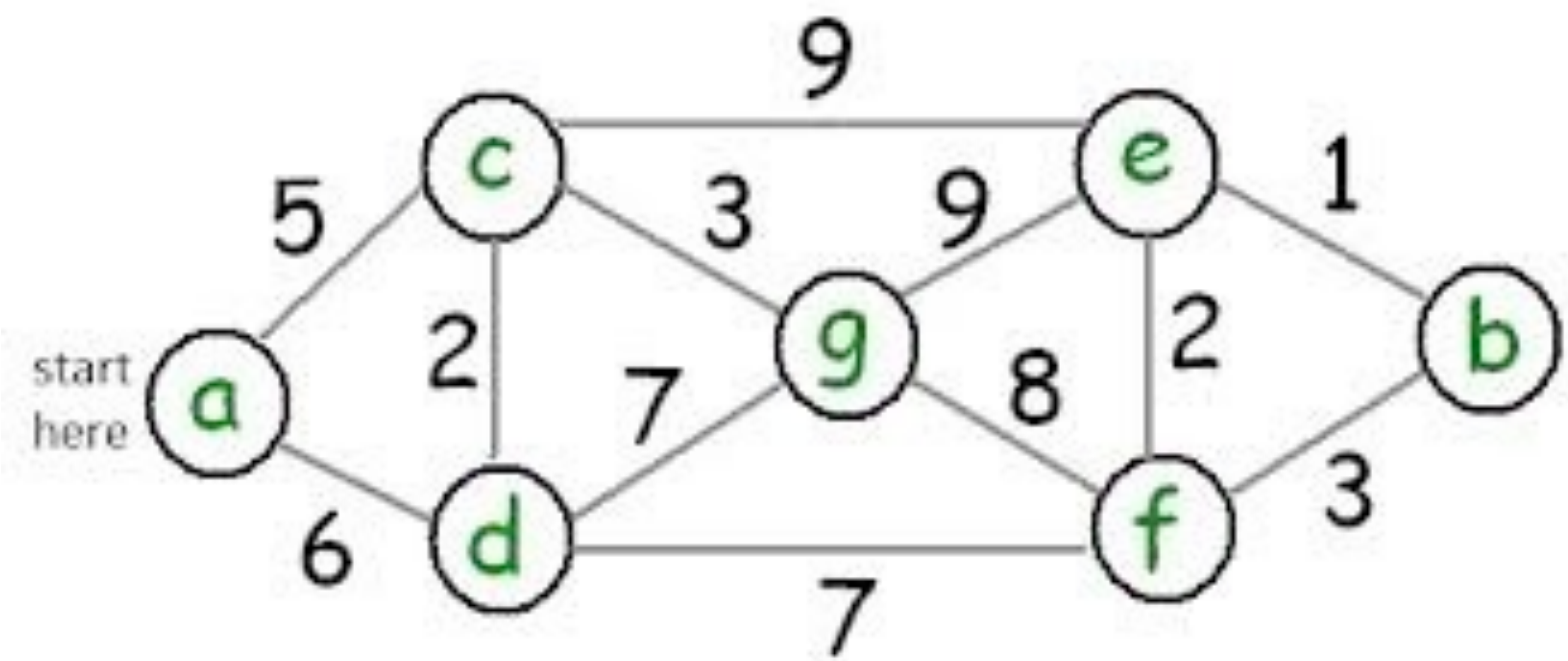
- (c) Given the binary heap below, show the resulting binary heap after a delete_min() operation.
  - remove 7
  - move 32 to root
  - 32 <==> 10
  - 32 <==> 12



Problem 3(a)

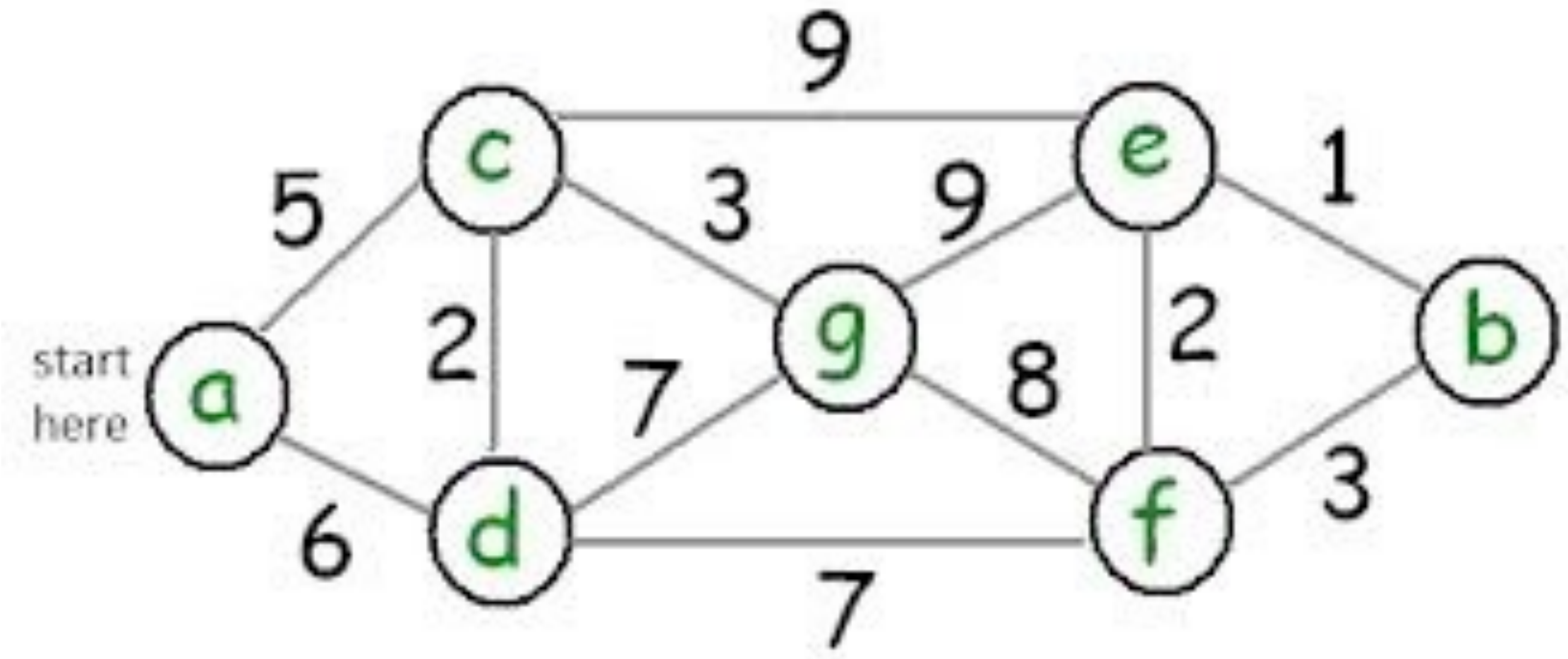
- Given the graph below and starting vertex a
  - (a) When using Prim's algorithm to find the minimum spanning tree of the graph, which edges are selected, in what order?

a -- c
c -- d
c -- g
d -- f
f -- e
e -- b



Problem 3(b)

- Given the graph below
  - (b) When using Dijkstra's algorithm to find the shortest paths from a, in which order are the vertices examined (via delete-min())? What is the shortest distance from a to each vertex?



Problem 3(b): Dijkstra's Algr.

	a	b	c	d	e	f	g
	0	∞	∞	∞	∞	∞	∞
a	0	∞	5	6	∞	∞	∞
c	0	∞	5	6	14	∞	8
d	0	∞	5	6	14	13	8
g	0	∞	5	6	14	13	8
f	0	16	5	6	14	13	8
e	0	15	5	6	14	13	8
b	0	15	5	6	14	13	8

Problem 4

- Given a list of n positive integers [v_1, v_2, ..., v_n], our goal is to maximize the sum of the integers we pick subject to the constraint that no two adjacent integers in the list can be picked. For example, given [5, 1, 2, 10, 6, 2], the maximum sum is 17 by picking $v_1 = 5$, $v_4 = 10$, $v_6 = 2$. Design a dynamic programming algorithm to solve this problem. What is the time complexity of your algorithm?

Problem 4

- $S(i)$: maximal sum when picking from $(v_1, v_2, \ldots, v_i)$
- Two cases when considering v_i
  - picking v_i or not picking v_i
  - $S(i) = \max\{S(i-2) + v_i, S(i-1)\}$
- Time complexity
  - n subproblems, each has two cases
  - $O(n)$