

Homework Set #1
Due Lecture 4
at the beginning of class

READINGS in AMO.

- Chapter 1
- Sections 2.1 to 2.3
- Pages 53 to 63.
- Pages 765 to 773.

HOMEWORK EXERCISES.**Notes:**

- As mentioned in class, and on the web site, groups of two may collaborate and hand in a single homework set. If you want to collaborate, but want help in identifying a partner, please let Professor Orlin know.
- Some of the exercises are odd numbered, and are available from Professor Orlin's home page (<http://web.mit.edu/jorlin/www/>). You are on your honor to give the odd numbered exercises a sincere effort before consulting the web site, and to make sure that any write-up is fully your own. Please do not use phrasing taken directly from the web site. The same rule applies to any other problem to which you may have access to a solution. If you consulted another solution in order to solve the exercise, please mention this in your write-up.
- Homework will be graded out of 3 points, as described on the web site. The grader will fully grade some exercises, and briefly scan others.
- This homework set, which has 12 exercises and an extra credit) is longer than the typical homework set for 15.082J/6855J.

Chapter 1 of AMO

1. Exercise 1.1.
2. Exercise 1.7. Also, explain how to interpret a solution to the shortest path as a solution to the paragraph problem.

Chapter 2 of AMO

3. Exercise 2.3
4. Exercise 2.13
5. Exercise 2.14
6. Exercise 2.30.

7. Exercise 2.31. (You can specify the forward star as in the book, or the adjacency lists as in the class notes.)
8. Exercise 2.50.

Chapter 3 of AMO

9. Exercise 3.3
10. Exercise 3.4
11. Exercise 3.6
12. Exercise 3.10.

Extra Credit

Repeat exercise 3.10 but show that the first-fit heuristic can be implemented to run in $O(n \log n)$ time using a variation of the leaf-heap data structure described in class. For each bin j , let $\text{key}(j)$ be the amount of space used up in bin j . Each parent node stores the minimum key of its two children. The update of the leaf-heap after inserting item k into a bin is easily accomplished in $O(\log n)$ steps. So, a solution to this exercise is pseudo-code for finding the first bin in which item k would fit, and a justification as to why this operation takes $O(\log n)$ steps.