

---

Ingredients: Sorting with a drop of amortization.

---

### Exercise Session

1. Discuss <http://java.sun.com/javase/6/docs/api/java/util/ArrayList.html>
  - What is the guarantee on the running time of `add`?
  - Is this better or worse than for `LinkedList`?
  - What does it mean that the "constant factor is low compared to `LinkedList`"?
  - How is this implemented?
  - What is the initial capacity you would choose? What does this parameter mean?
2. GT Ex. R-4.16 p. 252
3. GT C-4.5 p. 252
4. GT C-4.9 p. 253
5. GT C-4.13 p. 253

### Problems For Self-Study

1. GT Ex. R-4.14 p. 252 (ignore bubble-sort).
2. GT Ex. C-4.14 p. 253 *Hint*: read about RADIX-SORT
3. GT Ex. C-4.15 p. 253

### Homework Assignment

1. What operation is required by Dijkstra and A\*, but not supported by `java.util.PriorityQueue` (with sufficient speed)? Explain briefly how would you implement this operation in a binary heap. [2 hours]
2. GT Ex. C-4.16 p. 253 [1 hour]

**Deadline:** before the lecture on March 27.