CS 240: Data Structures and Data Management **Spring 2026**

Tutorial 02: May 29

1. **Recurrences** Below is a specification of an algorithm ALGO. It takes as input an array A of size n and indices $0 \leq i, j < n$. Provide a O bound for the runtime of ALGO$(A, 0, n-1)$ in terms of n. You may assume that n is a power of 3.

Algorithm 1: ALGO(A, i, j)

```
if A[j] < A[i] then
    swap A[j] and A[i]
end
if j − i + 1 > 2 then
    t ← ⌊(j−i+1)/3⌋
    ALGO(A, i, j − t)
    ALGO(A, i + t, j)
    ALGO(A, i, j − t)
end
```

Bonus: Prove that, ALGO$(A, 0, n-1)$ terminates, and that upon termination, A is sorted.

2. **Amortized analysis**

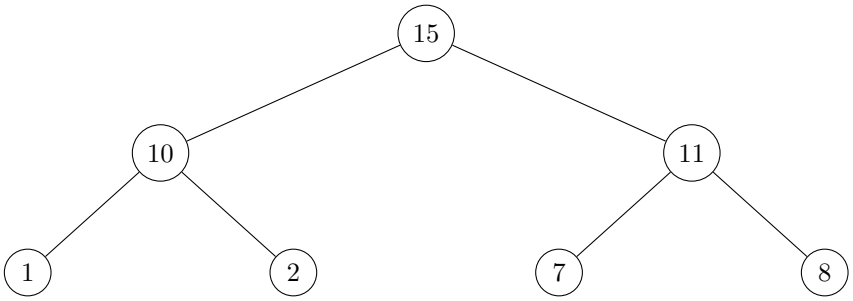
   (a) Simulate insertion into a dynamic array as follows:
   - Start with an array of size 1.
   - Insert the numbers $3264, 1728, 240, 65537$.[1]
   - Every time the array becomes full:
     - Create a new array with double the capacity.
     - Copy all existing items to the new array.
     - Continue the insertion.

   Write out the full array after each insertion:
   - when resizing happens,
   - how many items are copied, and
   - the total number of operations.

   (b) Consider now the general case. Beginning with an array of size 1, our goal is to insert n elements.
   (i) How many total insertions are performed?
   (ii) How many total copies are performed?
   (iii) Give a big-O bound for the sum of your answers to parts (i) and (ii).
   (iv) What can you conclude about the amortized cost of insertion into a dynamic array?

3. **Max-heap operations** Insert 27 and 9 into the following heap, then perform a delete-max operation on the resulting heap.

[1] Do these numbers mean anything to you?



4. **Heap applications** How would you implement a stack using a heap? Analyse the complexity of the push and pop operations.

5. **More heap applications** Suppose you are given an unsorted array A containing n integers and an integer k with $1 \leq k \leq n$. Design and analyse the runtime of an algorithm which outputs the k largest elements of A in any order.

   Bonus: Suppose you are given as input a max heap A containing n integers and an integer k with $1 \leq k \leq n$. Design an algorithm which outputs the k largest elements of A in time $O(k \log k)$.