CS 457, Fall 2019

Drexel University, Department of Computer Science

Lecture 10

Midterm

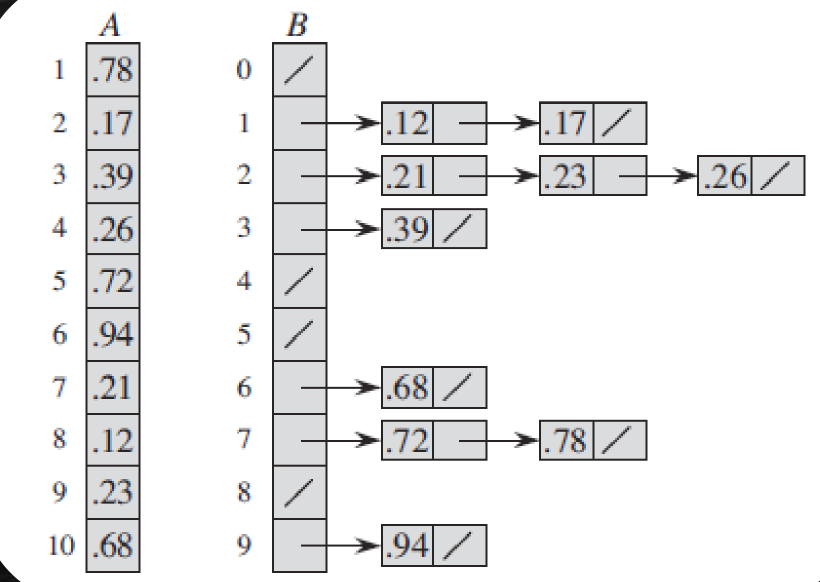
- In-class midterm on Wednesday
- You can bring a handwritten cheat-sheet
- Tested material can be from anything that we covered so far:
  - Chapters: 1, 2, 3, 4 (except 4.2 and 4.6), 5 (except 5.3), 7, 8 (only 8.4), 9, and 11 (only 11.1 and 11.2)
  - You can disregard the formal proofs of correctness that appear in some of these sections
- Preparation:
  - All of the homework problems and solutions closely
  - All of the problems discussed during the recitation
  - Other problems from the textbook

Sorting in Linear Time

- Assume that the input is drawn from uniform distribution from interval $[0, 1)$
- Can we use this information to get a faster algorithm, in the worst case sense?
- Can we use this information to get a faster algorithm, in the average case sense?

BUCKET-SORT(A)

let $B[0\,..\boxed{n-1}]$ be a new array
$n = A.length$
for $i = 0$ **to** $n-1$
    make $B[i]$ an empty list
for $i = 1$ **to** n
    insert $A[i]$ into list $B[\lfloor nA[i] \rfloor]$
for $i = 0$ **to** $n-1$
    sort list $B[i]$ with insertion sort
concatenate the lists $B[0], B[1], \ldots, B[n-1]$ together in order



Bucket Sort (Running Time)

- Let n_i be the number of elements in $B[i]$ (random variable)
- What is the running time, $T(n)$ of bucket sort as a function of n_i?
  - $T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$
- If we let $X_{ij} = \mathbb{I}\{A[j] \text{ falls in bucket } i\}$, then $n_i = \sum_{j=1}^{n} X_{ij}$

$$
\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \quad \text{(by linearity of expectation)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O\left(\mathrm{E}\left[n_i^2\right]\right) \quad \text{(by equation (C.22))} .
\end{aligned}
$$

Bucket Sort (Running Time)

- Let n_i be the number of elements in $B[i]$ (random variable)
- What is the running time, $T(n)$ of bucket sort as a function of n_i?
  - $T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$
- If we let $X_{ij} = \mathbb{I}\{A[j] \text{ falls in bucket } i\}$, then $n_i = \sum_{j=1}^{n} X_{ij}$

$$
\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \quad \text{(by linearity of expectation)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O\left(\mathrm{E}\left[n_i^2\right]\right) \quad \text{(by equation (C.22))} \ .
\end{aligned}
$$

It suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$

Bucket Sort (Running Time)

- So, it suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$
- What is the value of $\mathrm{E}[n_i^2]$?

$$
\begin{aligned}
\mathrm{E}\left[n_i^2\right] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n}\mathrm{E}\left[X_{ij}^2\right] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} \mathrm{E}\left[X_{ij}X_{ik}\right] ,
\end{aligned}
$$

Bucket Sort (Running Time)

- So, it suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$
- What is the value of $\mathrm{E}[n_i^2]$?

$$\begin{aligned}
\mathrm{E}\left[n_i^2\right] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n}\mathrm{E}\left[X_{ij}^2\right] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} \mathrm{E}\left[X_{ij}X_{ik}\right] ,
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}\left[X_{ij}^2\right] &= 1^2\cdot\frac{1}{n} + 0^2\cdot\left(1-\frac{1}{n}\right) \\
&= \frac{1}{n} .
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}\left[X_{ij}X_{ik}\right] &= \mathrm{E}\left[X_{ij}\right]\mathrm{E}\left[X_{ik}\right] \\
&= \frac{1}{n}\cdot\frac{1}{n} \\
&= \frac{1}{n^2} .
\end{aligned}$$

Simple Indicator Variables Example

Randomized-Loops (n)

$c = 0$
for $x = 1$ **to** n
    $k = \text{Random}(1, n)$ // Random number from 1 to n
    for $y = 1$ **to** k
        $c = c + 1$
return c

- What is the expected running time of Randomized-Loops(n)?
  - Let $X_{ij} = \mathbb{I}\{\text{The value of } k \text{ in the } i\text{–th outer loop iteration is equal to } j\}$
  - Then we wish to compute expected value of $X = \sum_{i=1}^{n} \sum_{j=1}^{n} j\, X_{ij}$
  - Taking the expectation, we get

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{i=1}^{n} \sum_{j=1}^{n} j\, X_{ij}\right] = \sum_{i=1}^{n} \sum_{j=1}^{n} j\, \mathrm{E}[X_{ij}] = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{j}{n} = \sum_{i=1}^{n} \frac{n+1}{2} = \frac{n(n+1)}{2}$$

Today’s Lecture

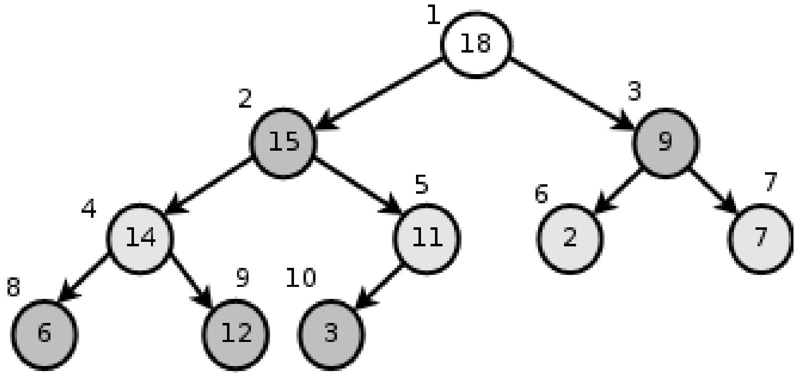
- Binary trees
  - Heaps and search trees
  - Structural induction

Binary Heaps

What is a binary heap? How is it stored in memory?

Heap data structure:

PARENT (i)
    return $\lfloor i/2 \rfloor$
LEFT (i)
    return $2i$
RIGHT (i)
    return $2i + 1$



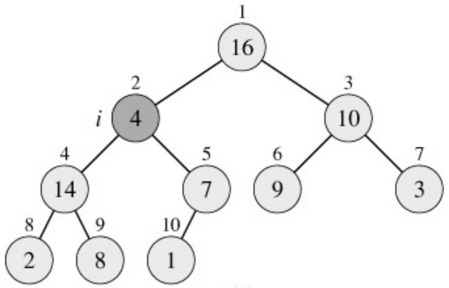
Max-heap property: A[PARENT(i)] $\geq$ A[i]

Heapsort

Max-Heapify (A, *i*)

```
l = left (i)
r = right (i)
if  l ≤ A.heap-size and A[l] > A[i]
        largest = l
else largest = i
if r ≤ A.heap-size and A[r] > A[largest]
        largest = r
if largest  ≠ i
        exchange A[i] with A[largest]
        Max-Heapify (A, largest)
```

Call to **Max-Heapify (A,2)**

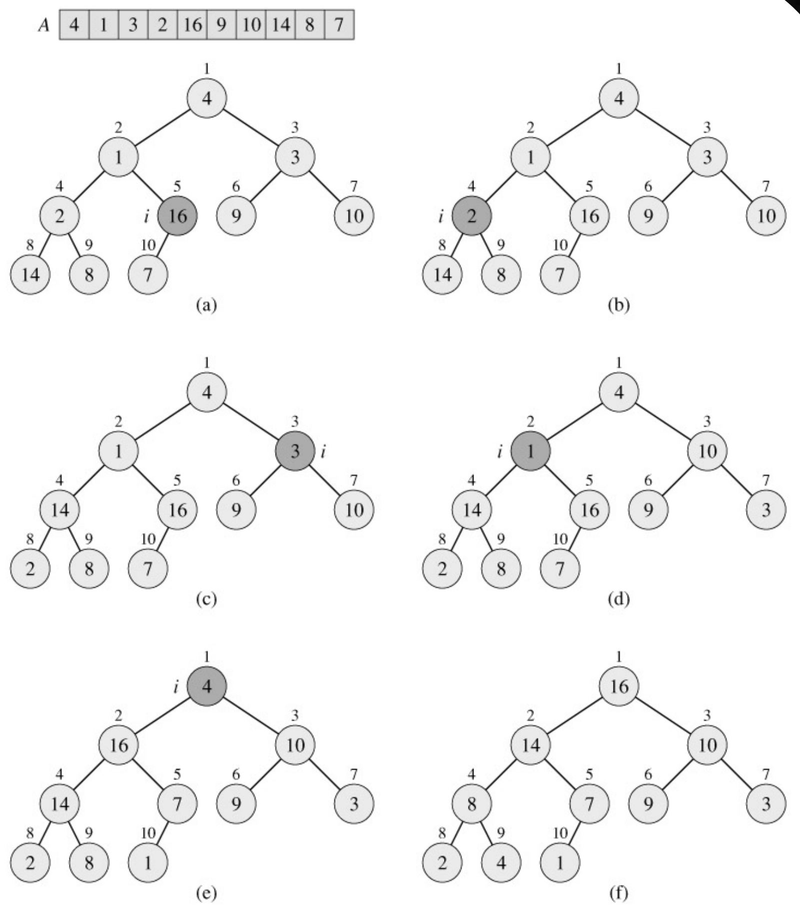


(a)

Heapsort

Build-Max-Heap (A)

1. A.heap-size = A.length
2. **for** $i = \lfloor \text{A.length}/2 \rfloor$ **down to** 1
3. Max-Heapify(A, i)



Heapsort

Heapsort(A)

```
Build-Max-Heap(A)
for i =A.length down to 2
    exchange A[1] with A[i]
    A.heap-size = A.heap-size -1
    Max-Heapify(A,1)
```