# Data Structures and Algorithms — Block 1, 2018
# Final Exam - Resit

December 11, 2018

## Problem 1 (25pt)

In each of the following question, please specify if the statement is true or false. If the statement is true, explain why it is true. If it is false, explain what the correct answer is and why. (For each question, 2 points for the true/false answer and 3 points for the explanations.)

(a) (5pt) $n^{1.5} = O(n \lg n)$
Answer: False. Assume that $n^{1.5} = O(n \lg n)$ i.e. there exist positive constants $n_0$ and $c$ such that $n^{1.5} \leq c(n \lg n)$ for all $n \geq n_0$, then we can get $\frac{n^{1.5}}{n \lg n} \leq c$. But there is no constant greater than $\frac{n^{1.5}}{n \lg n}$. So the assumption leads to contradiction.

(b) (5pt) A complete binary tree with depth 3 has at most 7 nodes.
Answer: False. At most 15 nodes.

(c) (5pt) $\sum_{i=1}^{n} 2i^3 = O(n^4)$
Answer: True. $\sum_{i=1}^{n} 2i^3 \leq \sum_{i=1}^{n} 2n^3 = n(2n^3) = 2n^4 = O(n^4)$.

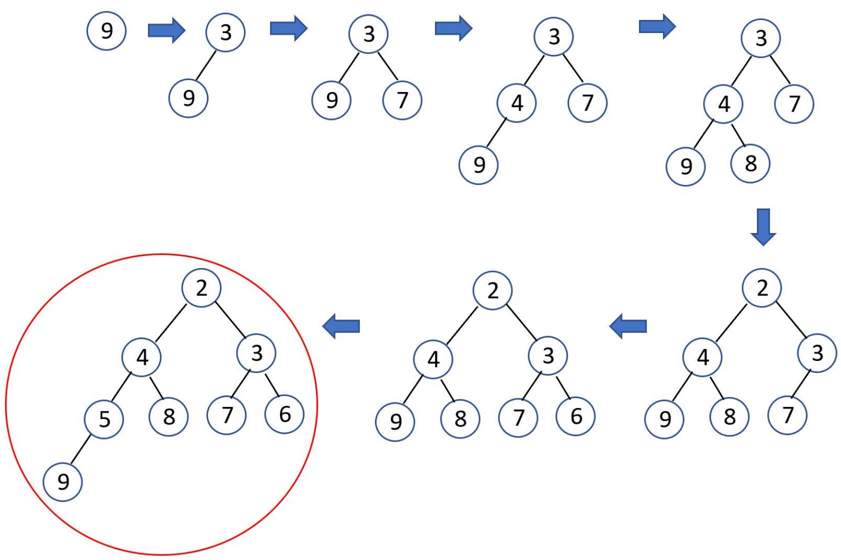
(d) (5pt) A chained hash table of size 64 can contain at most 64 elements.
Answer: False. It can contain unlimited number of elements.

(e) (5pt) $T(n) = 2T(n/4) + \Theta(\sqrt{n}) = \Omega(n^2)$.
Answer: False. Case 2 of the master theorem $\Longrightarrow T(n) = \Theta(\sqrt{n} lgn) = o(n^2) \Longrightarrow T(n) \neq \Omega(n^2)$.

## Problem 2 (10pt)

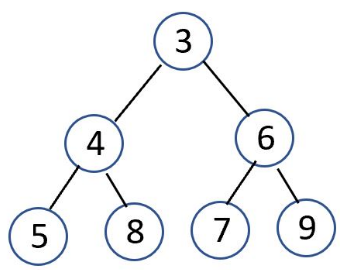
(a) (6pt) Draw the binary min-heap that results from inserting 9, 3, 7, 4, 8, 2, 6, 5 in that order into an initially empty binary min heap. You do not need to show the array representation of the heap. You are only required to show the final tree, although drawing intermediate trees may result in partial credit. If you draw intermediate trees, please circle your final result for any credit.

Answer:



(b) (4pt) Draw the result of one extract-min call on your heap draw in part (a).

Answer:




## Problem 3 (8pt)

Fill in the diagram below with the result of partitioning the array with standard quicksort partitioning (taking the E at the left as the partitioning item)(6pt). Also give the number of exchanges (2pt).

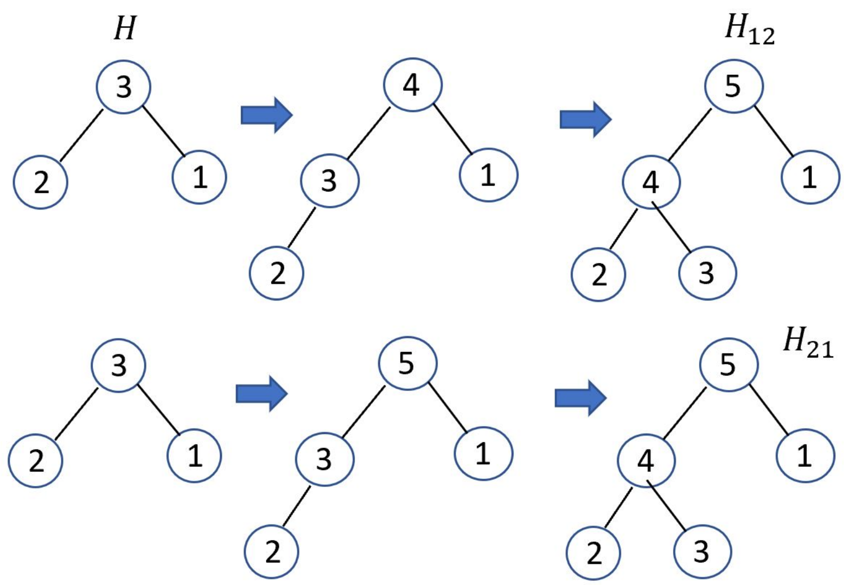
| E | V | E | R | Y | Q | U | A | L | K | E | Y | S | T | O | P | S | I | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | |

Answer: E E A E Y R Q U E L K V Y S T O P S I T, 4 exchanges

## Problem 4 (10pt)

Suppose we have a max-heap $H$ and two values $v_1$ and $v_2$, such that all values are distinct. Let $H_{12}$ be the heap you get if you insert $v_1$ and then $v_2$ into $H$, and $H_{21}$ be the heap you get if you insert $v_2$ and then $v_1$ into $H$. Give an example of $H$, $v_1$ and $v_2$ such that $H_{12} = H_{21}$. No justification needed, just draw the heaps $H$, $H_{12}$ and $H_{21}$.

Answer: $v_1 = 4$, $v_2 = 5$.




## Problem 5 (20pt)

Fill in this table with the worst-case asymptotic running time of each operation when using the data structure listed.

| | insert | search | delete | getMin | getMax |
|---|---|---|---|---|---|
| sorted array | $O(n)$ | $O(\lg n)$ | $O(n)$ | $O(1)$ | $O(1)$ |
| unsorted array | $O(1)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |
| array kept organized as a min-heap | $O(\lg n)$ | $O(n)$ | $O(n)$ | $O(1)$ | $O(n)$ |
| hash table with chaining for collision resolution | $O(1)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |

# Problem 6 (12pt)

Draw the contents of the two open addressing hash tables in the boxes below. The size of the hash table is 9. The hash function used is $h(k) = k \mod 9$.

(a) (5pt) What values will be in the hash table after the following sequence of insertions: 18, 16, 10, 7, 26 ? Assume that the probing constants (e.g. the $c$ in $h'(k) = h(k) + ci$ for linear probing, and the equivalent two constants $c_1, c_2$ for quadratic probing) are all 1. Draw the values in the boxes below, and show your work for partial credit.

Linear probing

| | |
|---|---|
| 0 | 18 |
| 1 | 10 |
| 2 | 26 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | 16 |
| 8 | 7 |

Quadratic probing

| | |
|---|---|
| 0 | 18 |
| 1 | 10 |
| 2 | |
| 3 | |
| 4 | 7 |
| 5 | |
| 6 | |
| 7 | 16 |
| 8 | 26 |

(b) (3pt) What is the load factor for the tables? Answer: $\alpha = \frac{5}{9}$.

(c) (2pt) The table with linear probing will (circle one):

(i) gradually degrade in performance as more values are inserted

(ii) may fail to find a location on the next insertion

(iii) none of the above

Answer: (i)

(d) (2pt) The table with quadratic probing will (circle one):

(i) gradually degrade in performance as more values are inserted

(ii) may fail to find a location on the next insertion

(iii) none of the above

Answer: (ii)

# Problem 7 (15pt (+ 5 bonus))

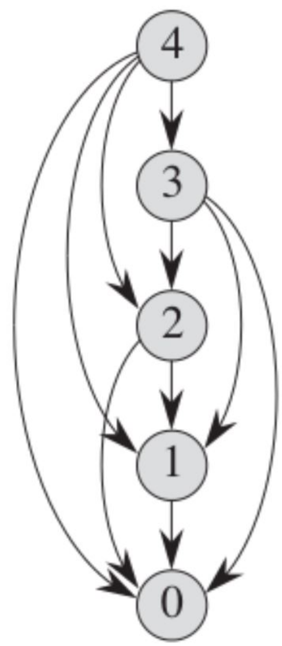
Remember the rod cutting problem from the lecture: Given a rod of length $n$, and list of prices for length $i$, where $1 \leq i \leq n$, how should you cut the rod into pieces in order to maximize the revenue you can get?

(a) (10pt + bonus 5pt) Write down the bottom-up dynamic programming algorithm that solves the problem. The algorithm should return an array $r$ that contains the size of each rod piece. You can either do it in pseudocode or Python. If you write it in Python, you get bonus 5 points, but it has to run correctly. A Python algorithm that is partially correct will be considered pseudocode and will get you no bonus points.

Answer:

```
BOTTOM-UP-CUT-ROD(p,n)
    let r[0..n] be a new array
    r[0]=0
    for j=1 to n
        q=-\infty
        for i=1 to j
            q=max(q,p[i]+r[j-i])
        r[j]=q
    return r[n]
```

(b) (5pt) Draw the subproblem graph for a rod of size 4. How many vertices and edges are in the graph?
Answer: 5 vertices, 10 edges.


4
3
2
1
0