Algorithms
Prof. Rodolà
2^{nd} Semester 2019/2020

Bioinformatics
Sapienza University of Rome

Midterm self-evaluation

Thu 23 Apr 2020

Solutions

Question 1 (1 point).

Explain why the following statement is meaningless: "The running time of algorithm A is at least $O(n^2)$".

Let $T(n)$ be the running time of algorithm A. The statement "$T(n)$ is at least $O(n^2)$" means $T(n) \geq f(n)$ for some function $f(n) = O(n^2)$, so it does not tell us anything about the *upper bound* of $T(n)$. At the same time, $f(n) = O(n^2)$ means that $f(n)$ could be any fuction asymptotically smaller than n^2, for example $f(n) = n$. In other words, we can not conclude anything about the *lower bound* of $T(n)$ either.

Question 2 (2 points).

Is $2^{n+1} = O(2^n)$? Why yes/no?

Is $2^{2n} = O(2^n)$? Why yes/no?

Yes, because $2^{n+1} = 2 \times 2^n$. So we can choose $c \geq 0$ and $n_0 = 0$ s.t. $0 \leq 2^{n+1} \leq c \times 2^n$ for all $n \geq n_0$.

No, because $2^{2n} = 2^n \times 2^n = 4^n$. Therefore we can not find any c, n_0 s.t. $0 \leq 4^n \leq c \times 2^n$ for all $n \geq n_0$.

Question 3 (4 points).

Using the substitution method, show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

Since we need to prove that is $T(n) = O(n^2)$ let us guess $T(n) \leq cn^2$.

Applying the substitution method, we get:

$$\begin{aligned} T(n) &\leq c(n-1)^2 + n \\ &= cn^2 - 2cn + c + n \\ &= cn^2 + n(1-2c) + c \\ &\leq cn^2 \end{aligned}$$

Question 4 (6 points)**.**

Consider the recursion tree for $T(n) = 3T(\lfloor n/2 \rfloor) + n$.

- What is the subproblem size for a node at depth i?
- How many levels are in the tree?
- How many leaves are in the tree?
- What is the total cost over all nodes at depth i (excluding the leaf level)?
- Using your answers to the previous points, provide an asymptotic upper bound for the recursion.

Answers:

- $\frac{n}{2^i}$
- $\lg n$
- $3^{\lg n} = n^{\lg 3}$
- For $i = 0 \ldots \lg n - 1$, the cost is $3^i \frac{n}{2^i} = \left(\frac{3}{2}\right)^i n$
-

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\lg n - 1} \left(\frac{3}{2}\right)^i n + \Theta(n^{\lg 3}) = n \sum_{i=0}^{\lg n - 1} \left(\frac{3}{2}\right)^i + \Theta(n^{\lg 3}) \\
&= 2n \left(\left(\frac{3}{2}\right)^{\lg n} - 1 \right) + \Theta(n^{\lg 3}) = 2(n^{\lg 3} - n) + \Theta(n^{\lg 3}) \\
&= O(n^{\lg 3})
\end{aligned}$$

Question 5 (3 points)**.**

Is an array that is in sorted order a min-heap?

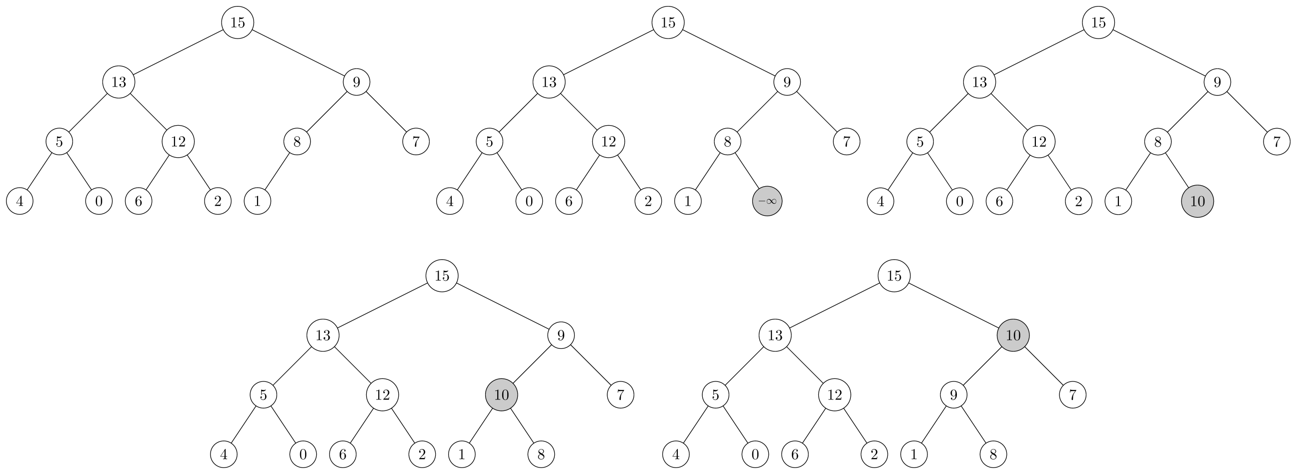
Is the array with values $\langle 23, 17, 14, 6, 13, 10, 1, 5, 7, 12 \rangle$ a max-heap?

Yes, because for any index i we have that LEFT(i) and RIGHT(i) are larger and the elements indexed by them are $\geq A[i]$.

No, because in this array PARENT(7) $= 6$, which violates the max-heap property.

Question 6 (3 points)**.**

Illustrate the operation of MAX-HEAP-INSERT$(A, 10)$ on the heap $A = \langle 15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1 \rangle$.



Question 7 (3 points)**.**

Illustrate the result of each operation in the sequence ENQUEUE$(Q, 4)$, ENQUEUE$(Q, 1)$, ENQUEUE$(Q, 3)$, DEQUEUE(Q), ENQUEUE$(Q, 8)$, and DEQUEUE(Q) on an initially empty queue Q stored in array $Q[1..6]$.

ENQUEUE$(Q, 4)$	4			
ENQUEUE$(Q, 1)$	4	1		
ENQUEUE$(Q, 3)$	4	1	3	
DEQUEUE(Q)		1	3	
ENQUEUE$(Q, 8)$		1	3	8
DEQUEUE(Q)			3	8

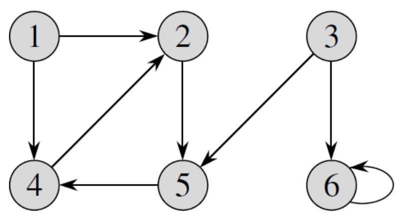
Question 8 (2 points).

Write an $O(n)$-time recursive procedure that, given an n-node binary tree, prints out the key of each node in the tree.

```
PRINT–BINARY–TREE(T)
    x = T.root
    if x != NIL
        PRINT–BINARY–TREE(x.left)
        print x.key
        PRINT–BINARY–TREE(x.right)
```

Question 9 (2 points).

For the directed graph in the figure, specify for each node the d and π values that result from running breadth-first search using vertex 3 as the source.



vertex	1	2	3	4	5	6
d	∞	∞	0	∞	∞	∞
π	NIL	NIL	NIL	NIL	NIL	NIL

vertex	1	2	3	4	5	6
d	∞	∞	0	∞	1	1
π	NIL	NIL	NIL	NIL	3	3

vertex	1	2	3	4	5	6
d	∞	∞	0	2	1	1
π	NIL	NIL	NIL	5	3	3

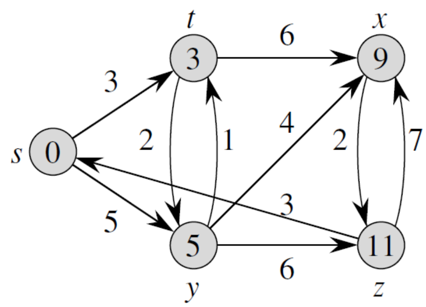
vertex	1	2	3	4	5	6
d	∞	3	0	2	1	1
π	NIL	4	NIL	5	3	3

Vertex 1 is not reachable from vertex 3.

Question 10 (6 points)**.**

Run Dijkstra's algorithm on the directed graph below.

- Do it twice, with two different sources: first s, and then z.
- For each iteration of the while loop, show the d and π values for each node.



From source s:

d values:

s	t	x	y	z
0	3	∞	5	∞
0	3	9	5	∞
0	3	9	5	11
0	3	9	5	11
0	3	9	5	11

π values:

s	t	x	y	z
NIL	s	NIL	NIL	NIL
NIL	s	t	s	NIL
NIL	s	t	s	y
NIL	s	t	s	y
NIL	s	t	s	y

From source z:

d values:

s	t	x	y	z
3	∞	7	∞	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0

π values:

s	t	x	y	z
z	NIL	z	NIL	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL