CSOR W4231 Analysis of algorithms I
Midterm

Xijiao Li (xl2950)

September 29, 2020

Problem 1

a.

Use Master Theorem. $a = 4, b = 2, f(n) = n^{1.5}$
because $n^{\log_b a} = n^{\log_2 4} = n^2, f(n) = \Theta(n^2)$, case 1 of the Master Theorem
applies and we have $T(n) = \Theta(n^{\log_b a}) = O(n^2)$

b.

Unfold the recurrences:

$$
\begin{aligned}
T(n) &= T(n-1) + \sqrt{n} \\
&= T(n-2) + \sqrt{n-1} + \sqrt{n} \\
&\cdots \\
&= T(1) + \sqrt{2} + \ldots + \sqrt{n} \\
&= T(1) + \sum_{k=2}^{n} \sqrt{k} \\
&= T(1) + H_n^{-1/2} - 1
\end{aligned}
$$

Since $\sqrt{k}$ is monotonically increasing, we use Approximation by integrals formula to approximate the sum:

$$
\sum_{k=2}^{n} \sqrt{k} \leq \int_{2}^{n+1} \sqrt{k} dx
$$

So $T(n) = O(n^{3/2})$

1

Problem 2

1.

$f = \Theta(g)$

2.

$f = o(g)$

3.

$f = \Theta(g)$

4.

$f = \omega(g)$

3.

1.

True. $T(n) = 2T(n/2) + o(n)$, use Master Therom we know its $n \log n$.

2.

True. If converting the Max Heap into a Tree, the minimum element must be found at a leaf nodes since every node in this tree is less than its parent. In the worst case, all leaf nodes are full, so the height of the tree is $\log n - 1$ and there are $2^{\log n - 1} = n/2$ leaf nodes, which has indices $\lceil \frac{n}{2} \rceil...n$ in the heap.

3.

False. The 8 and 4 must appear at the right and left side of 6 respectively, instead of just following the 6.

4.

True. We can first duplicate the array while exchanging each element in a pair for any element in this list. Then we can sort these two lists of pairs by converting each pair to a number with the form $in + j$, so that they will be two lists of number in the range 0 to n^2, and then we can use the Radix Sort here as the problem in sample midterm. After that, we just need to go through the pairs in the original array and see whether there is a same pair in the duplicated list. Since both lists are sorted so it takes O(n) time.

5.

True. Build a decision, so that it must has n leaf nodes so the height of this tree is $\log n$ and thus this model requires $\log n$ comparison.

4.

1. for all $(x_i x_{i-1})/(y_i - y_{i-1}) = c$ which means they are in one line.

2. we need to find the two median number of x and the two median numbers of y. If they are from two number x_i, y_i and x_j, y_j then it is partitional.

3. We can use linear time selection algorithm on x to find two median x_i and x_j and get y_i, y_j

We can then use linear time selection algorithm again to find two median ya and yb then check whether y_i, y_j is y_a, y_b, which is $O(n)$

5.

1. $O(1) + O(n) + O(n^2) = O(n^2)$
2. $\sum_1^{n-1} i/2 = n(n-1)/4$
3. use augmented search tree (told in lecture), which has $O(n \log n)$ since we need to insert and search.