HE Jianle
20536134
jhebg@ust.hk

COMP 3711 – Design and analysis of Algorithms
2019 Spring Semester – Written Assignment #2

Problem 1

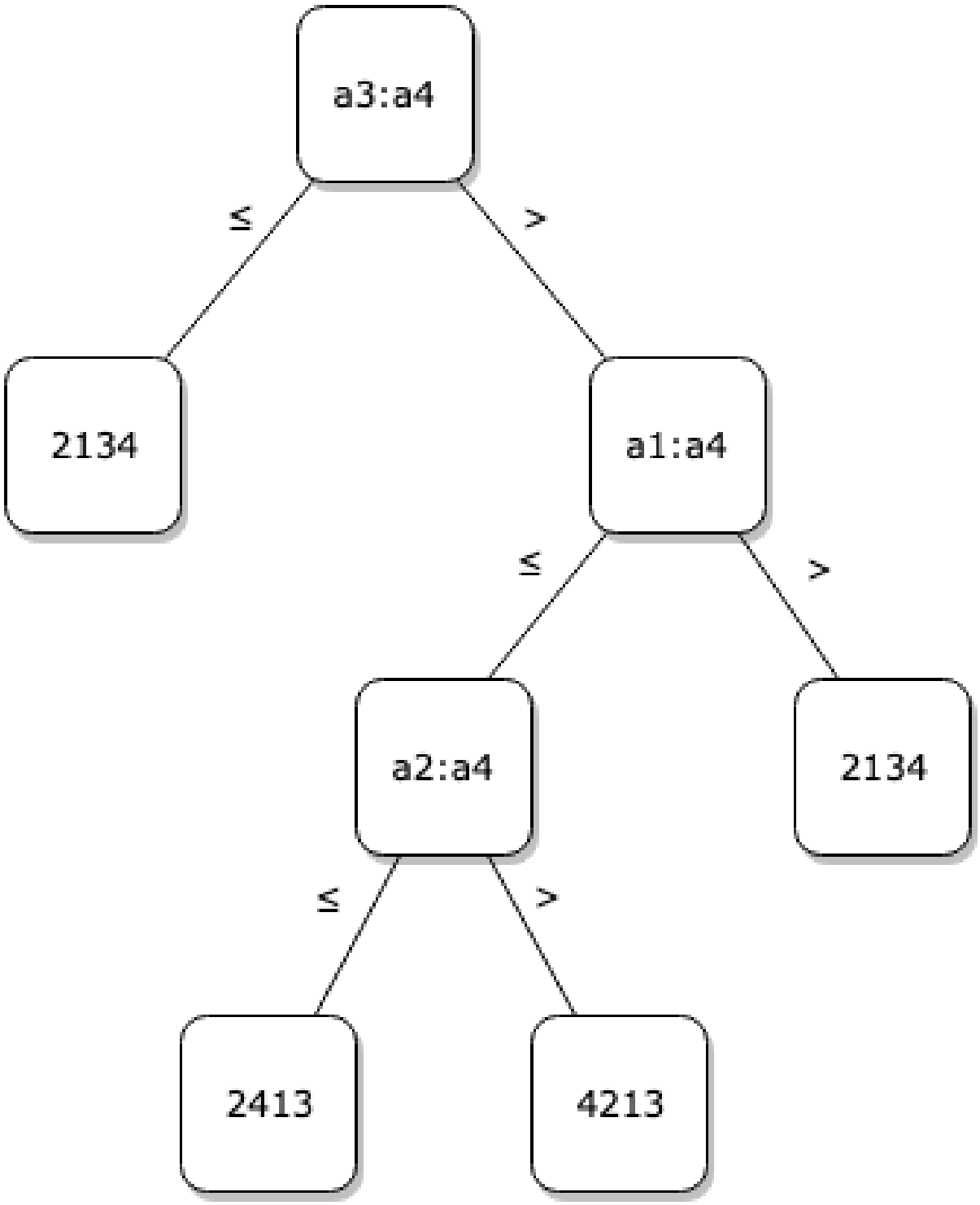
1. MEDIAN(A,p,r): using Quart()
   for (i <- 0 to $\left\lceil \frac{r-p+1}{4} \right\rceil - 1$)
        quart <- Quart(A,p,r-i)
        index <- LinearSearch(quart)
        SWAP(index, r-i)
   return Quart(A,p,r-$\left\lceil \frac{r-p+1}{4} \right\rceil$)

2. Taking MEDIAN(A,p,r) as the pivot
   Every time evenly divide the array into two parts.
   The recurrence will occur log n levels, and n comparison for each level. In total O(nlogn).

3. Using the modified Quicksort in (2).
   If k = n, it is what an ordinary quick sort does. The base case is only one element in the array.
   If k < n, there is an early termination. The base case will be $\left\lceil \frac{n}{k} \right\rceil$ elements in the array.

HE Jianle
20536134
jhebg@ust.hk

Problem 2



HE Jianle
20536134
jhebg@ust.hk

Problem 3
Reference: tutorial 5 question 7

Key the points in the array. Walk through the array as follows.
1. Set $x = x_1$
2. Walk through the points in increasing order.
   until finding the first j such that $x_j > x + 1$
3. Output
4. If there is no such j in (2), otherwise set $x = x_j$
5. Go to step (2)

Since each point is seen only once, this is an O(n) algorithm.

Proving correctness by induction:
If # of house = 1, it is correct.
If # of house < n, it is correct.
If # of house = n,
O(i,j) = minimum # of base stations need to cover $\{x_i,\ldots,x_j\}$
G(i,j) = minimum # of base stations Greedy uses to cover $\{x_i,\ldots,x_j\}$
Assume $[x_1, x_1 + 8]$ does not cover all of X because, if it did, Greedy would return that same one interval solution which is optimal.
Let j' be the smallest index such that $x_{j'} > x_1 + 8$
Greedy returns $[x_1, x_1 + 8]$ concatenated with the greedy solution for $\{x_j, \ldots, x_n\}$
=> Greedy uses 1 + O(j',n) intervals
By induction, Greedy solution for $\{x_j, \ldots, x_n\}$ is optimal.
Now, suppose that there is a solution to OPT different than the Greedy one.
Let [x,x + 8] be interval on OPT with the leftmost starting point.
$x \leq x1$ because otherwise x_1 would not be covered by any interval in OPT.
Let k be the minimum index such that $x_k > x + 8$
After removing [x,x + 8] remaining intervals in OPT must form optimal solution
for $\{x_k, \ldots, x_n\}$, otherwise we could build better solution using fewer base stations.
=> Optimal uses 1 + O(k,n) base station

Greedy uses 1 + O(j',n) base stations
Optimal uses 1 + O(k,n) base stations

$$\begin{aligned} G(1,n) &= 1 + G(j',n) \\ &= 1 + O(j',n) \\ &\leq 1 + O(k,n) \\ &= O(1,n) \end{aligned}$$

HE Jianle
20536134
jhebg@ust.hk

Problem 4

Reference: Method 2 in https://www.geeksforgeeks.org/median-of-two-sorted-arrays/

(a)

```
m_x <- x[⌊n/2⌋]
m_y <- x[⌊n/2⌋]
if n = 2
     return max(X[0],Y[0])
if m_x = m_y
     return m_x
if m_x > m_y
     X <- X[0 to ⌊n/2⌋]
     Y <- Y[⌊n/2⌋ to n-1]
if m_x < m_y
     X <- X[⌊n/2⌋ to n-1]
     Y <- Y[0 to ⌊n/2⌋]
n <- ⌊n/2⌋
MED(X,Y)
```

(b)

Every time finding the median of two arrays. If both of them are the same, it is the target output. Else the target output will fall into the range between two medians. I.e.: if $m_x > m_y$ then the target will fall into first half of X and second half of Y, and vice versa. Repeat the step until there are only two elements in the array. Pick 2^{th} (lower median) or 3^{rd} (upper median) out of 4 elements as the output.

(c)

$$
\begin{aligned}
T(1) &= 1 \\
T(n) &= 2T\left(\frac{n}{2}\right) + 1 \\
&= 2^2 T\left(\frac{n}{2^2}\right) + 2 \\
&= 2^3 T\left(\frac{n}{2^3}\right) + 3 \\
&\dots \\
&= 2^h T(1) + \log n \\
&= O(\log n)
\end{aligned}
$$