**CSE548, AMS542: Analysis of Algorithms, Spring 2021** **Date:** Apr 4

# Homework #2

**( Due: Apr 16 )**

**Task 1. [ 150 Points ] Average-Case Analysis of Median-of-3 Quickselect**

Consider the selection algorithm given in Figure 1.

Median-of-3-Quickselect( $A[q:r]$, $k$ )

**Input:** An array of distinct elements, and an integer $k \in [1, r-q+1]$.

**Output:** An element $x$ of $A[q:r]$ such that $rank(x, A[q,r]) = k$.

1. $n \leftarrow r - q + 1$
2. ***if*** $n < 1$ ***then return*** nil
3. ***elif*** $n < 4$ ***then***
4.     sort $A[q:r]$ in increasing order of value
5.     ***return*** $A[q+k-1]$
6. ***else***
7.     $x \leftarrow$ median of $A[q]$, $A[q+1]$ and $A[q+2]$
8.     rearrange the numbers of $A[q:r]$ such that
    - $A[t] = x$ for some $t \in [q, r]$,
    - $A[i] < x$ for each $i \in [q, t-1]$,
    - $A[i] > x$ for each $i \in [t+1, r]$,
9.     ***if*** $k = t - q + 1$ ***then return*** $A[t]$
10.     ***else if*** $k < t - q + 1$ ***then return*** Median-of-3-Quickselect( $A[q:t-1]$, $k$ )
11.         ***else return*** Median-of-3-Quickselect( $A[t+1:r]$, $k$ )

Figure 1: [Task 1] Return the $k^{th}$ smallest number in the input array $A[q:r]$, where $k \in [1, r-q+1]$.

Given an input array $A$ of size $n$, in this task we will analyze the average number of element comparisons (i.e., comparisons between two numbers of $A$) performed by this algorithm over all $n!$ possible permutations of the numbers in $A$ and all $n$ possible values of $k$. We will assume that the partitioning algorithm (in line 8) is *stable*, i.e., if two numbers $y$ and $z$ end up in the same partition and $y$ appears before $z$ in the input, then $y$ must also appear before $z$ in the resulting partition.

($a$) [ **15 Points** ] Show how to implement steps 7 and 8 of Figure 1 to get a stable partitioning of the numbers in $A[1:n]$ using only $n - \frac{1}{3}$ element comparisons on average, where the average is taken over all $n!$ possible permutations of the input numbers.

($b$) [ **25 Points** ] Let $t_n$ be the average number of element comparisons performed by the algorithm given in Figure 1 on an input array $A$ of size $n$, where $n \geq 0$ and the average is taken over all $n!$ possible permutations of the numbers in $A$ and all $n$ possible values of integer $k$. Argue that

$$t_n = \begin{cases} 0 & \text{if } n < 2, \\ 1 & \text{if } n = 2, \\ 3 & \text{if } n = 3, \\ n - \frac{1}{3} + \frac{6}{n^2(n-1)(n-2)} \sum_{k=1}^{n} (k-1)(n-k)((k-1)t_{k-1} + (n-k)t_{n-k}) & \text{otherwise.} \end{cases}$$

$(c)$ [ **50 Points** ] Let $T(z)$ be a generating function for $t_n$:

$$T(z) = t_0 + t_1 z + t_2 z^2 + \ldots + t_n z^n + \ldots \quad \ldots$$

Show that

$$z(1-z)^2\ T^{(4)}(z) + 3(1-z)^2\ T^{(3)}(z) - 12z\ T^{(2)}(z) - 12\ T^{(1)}(z) = \frac{8(z^2+8z+6)}{(1-z)^4},$$

where, $T^{(k)}(z) = \frac{d^k}{dz^k}(T(z))$ for integer $k > 0$.

$(d)$ [ **25 Points** ] Use the result from part $(c)$ to show that

$$z(1-z)^2\ T^{(3)}(z) + 2(1-z)\ T^{(2)}(z) + 2(1-6z)\ T^{(1)}(z) = \frac{8(z^2+3z+1)}{(1-z)^3} - 4.$$

$(e)$ [ **25 Points** ] Use the result from part $(d)$ to show that

$$n^2(n-2)t_n = \frac{1}{2}(n+1)\left[(n-1)(n-6)t_{n-1} + n(n+1)t_{n+1}\right] - 10n(n-1) - 4.$$

$(f)$ [ **10 Points** ] What bound on $t_n$ can you get from part $(e)$?

COMP-S( $n$ ) $\{Integer\ n \geq 0\}$

***if*** $n = 0$ ***then return*** $1$
***else***
  $s \leftarrow$ COMP-S ( $n - 1$ )
  $t \leftarrow$ COMP-T ( $n - 1$ )
  ***return*** $s + 2t$

COMP-T( $n$ ) $\{Integer\ n \geq 0\}$

***if*** $n = 0$ ***then return*** $0$
***else***
  $s \leftarrow$ COMP-S ( $n$ )
  $t \leftarrow$ COMP-T ( $n - 1$ )
  ***return*** $3s + t$

Figure 2: [Task 2] Two mutually recursive functions.

**Task 2. [ 90 Points ] Two Mutually Recursive Functions**

Figure 2 shows two mutually recursive functions COMP-S and COMP-T. Both functions accept a nonnegative integer as the sole input.

Now answer the following questions.

($a$) [ **45 Points** ] Use generating functions to find the values returned by COMP-S( $n$ ) and COMP-T( $n$ ).

($b$) [ **45 Points** ] Use generating functions to find the running times of COMP-S( $n$ ) and COMP-T( $n$ ).

RANDOM-FLIPS( $A$, $p$, $q$ ) $\{$*Input is a bit array* $A[p:q]$*, where* $p, q \in \mathbb{Z}^+$*,* $q > p$*, and* $n = q - p + 1 = 2^k$ *for some integer* $k \geq 0$.$\}$

$n \leftarrow q - p + 1$
***if*** $n \leq 4$ ***then return***
***else***
  ***for*** $i \leftarrow 1$ ***to*** $n^{\log_2\left(\frac{\sqrt{61}-1}{6}\right)}$ ***do***
    $r \leftarrow$ RANDOM( $p$, $q$ ) $\{r \leftarrow$ *an integer chosen uniformly at random from* $[p, q]\}$
    $A[r] \leftarrow 1 - A[r]$
  $t \leftarrow$ RANDOM( $1$, $10$ )
  ***if*** $t$ is divisible by 2 ***then*** RANDOM-FLIPS( $A$, $p$, $q$ )
  ***elif*** $t$ is divisible by 3 ***then*** RANDOM-FLIPS( $A$, $p$, $p + \frac{n}{4} - 1$ )
  ***elif*** $t$ is divisible by 5 ***then*** RANDOM-FLIPS( $A$, $q - \frac{n}{4} + 1$, $q$ )
  ***elif*** $t$ is divisible by 7 ***then*** RANDOM-FLIPS( $A$, $p + \frac{n}{4}$, $q - \frac{n}{4}$ )
  ***else return***

Figure 3: [Task 3] Recursive random bit flips.

**Task 3. [ 60 Points ] A Recursive Random Bit Flipping Algorithm**

The recursive algorithm given in Figure 3 flips the entries of a bit array at random. Write down a recurrence relation describing the expected running time of the algorithm and solve it.