CSE 321 Introduction to Algorithm Design

Fall 2019

Name:Berke Süslü

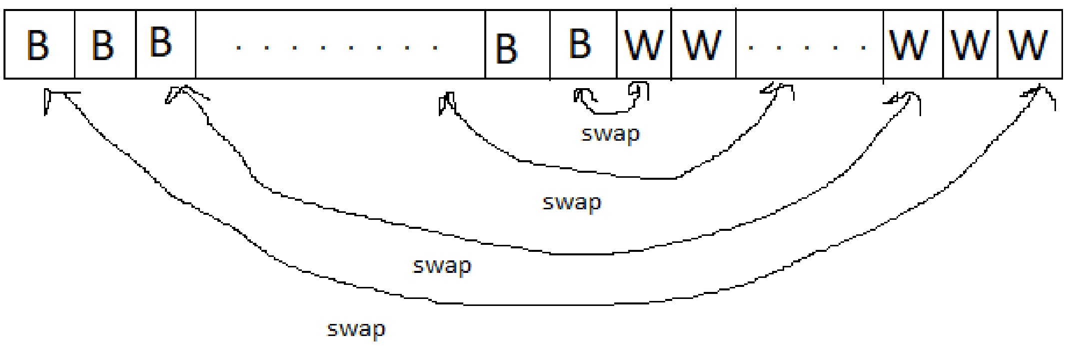
Number:161044076

1) Psuedocode black_and_white_boxes(array[0:2n],i)

```
if n>0
        swap(array[2n-i],array[i])
        black_and_white_boxes(array[0:2n],i-2)
end if
end
```

(Note: 2n is size of array and i is n)

The function swaps from middle to end points.



The complexity of T_{worst} , T_{best} , $T_{average}$ is same.

The recurrence relation: $T(n)=T(n-2)-O(1)$ => $O(1)$ is the swap function complexity.

$T(n)=T(n-4)-2*O(1)$

$T(n)=T(n-6)-3*O(1)$

. . . .

$T(n)=T(1)-(n/2)*O(1)$

$T(1)=0$ so, $T(n) = \Theta(n)$

2) Psuedocode find_fake_coin(array[0:n])

```
currentcoin = find_3_coin(array[0],array[1],array[2])
if currentcoin == false
        if n>= 3
                return find_fake_coin(array[3:n])
        else
                return find_fake_coin(array[n-3:n])
        end if
else
        return currentcoin
end if
```

end

Psuedocode find_3_coin(c1,c2,c3)

```
if c1 == c2 == c3
        return false
else if c1 == c2 & c2 != c3
        return c3
else if c1 == c3 & c1 != c2
        return c2
else
        return c1
end if
```

end

The complexity of T_{worst} , T_{best} , $T_{average}$ is same.

The recurrence relation: T(n)=T(n-3)-O(1) => O(1) is the complexity of find_3_coin

T(n)=T(n-6)-2*O(1)

T(n)=T(n-9)-3*O(1)

. . . .

T(n)=T(1)-(n/3)*O(1)

T(1)=0 so, T(n) = Θ(n)

3) Insertion Sort Average Case:

Step 1: Place A[2] into its proper position.
Step 2: Place A[3] into its proper position.
Step n-1:Place A[n] into its proper position.

Let T: be the number of basic operations at the step i, where 1≤i≤n-1

$T=T_1+T_2+...+T_{n-1}=\sum_{i=1}^{n-1} T_i$
$A(n)=E[T_1]+E[T_2]+...+E[T_{n-1}]=E[\sum_{i=1}^{n-1} T_i]$
$E[T_i]=\sum_{j=1}^{i} j * Prob(T_i = j)$

1 comparison will occur if x = A[i] > A[i-1]
2 comparison will occur if A[i-2]<x<A[i-1]
. . . .
i comparison will occur if A[1]<x<A[2]
i comparison will occur if x<A[1]

$P(T_i = j) = \begin{cases} \frac{1}{i+1} & \text{if } i\leq j\leq i-1 \\ \frac{2}{i+1} & \text{if } j=1 \end{cases}$

$E[T_i]=\sum_{j=1}^{i-1}(j * \frac{1}{i+1})+i*\frac{2}{i+1} = \frac{i}{2}+1-\frac{1}{i-1}$
$A(n)=E[T]= \sum_{i=1}^{n-1} E[T_i]$
$A(n)=\frac{n*(n-1)}{4}+n-1-\sum_{i=1}^{n-1} \frac{1}{i+1}$ → Harmonic Series
$A(n)= \Theta(n^2)$

Quicksort Average Case:

Assume that it is equally likely that the pivot element (A[left]) will be placed in any position after Partition.
$A(n)=E[T]=E[T_1]+E[T_2]$
T_1→# of operations in Partition. (fixed = n+1)
T_2→# of operations in recursive calls.
If proper position of the pivot is A[1]; then the 1 sublist are non-existed and A[2:n]
If proper position of the pivot is A[2]; then the 2 sublist are A[1:1] and A[3:n]
. . . .
If proper position of the pivot is A[i]; then the 2 sublist are A[1:i-1] and A[i+1:n]

E[T_2]=$\sum_x E[T_2 \mid X = x].P(X = x)$

P(X = x) = 1/n (due to uniform probability dist.)

A(n)=(n+1)+[A(0)+A(n-1) + A(1)+A(n-2)+ … + A(n-2)+A(1)+A(n-1)+A(0)]*$\frac{1}{n}$

A(n)=(n+1)+$\frac{2}{n}$*[A(0)+A(1)+…+A(n-1)] → Full history recurrence relation

n.A(n)=n*n(n+1)+2[A(0)+…+A(n-1)]

(n-1).A(n-1)=n.(n-1)+2[A(0)+…+A(n-2)]

Subtract operation

n.A(n)-(n-1)*A(n-1)=2n+2.A(n-1)

Assume t(n)=$\frac{A(n)}{n+1}$

t(n)=t(n-1)+$\frac{2}{n+1}$

A(0)=0 → initial condition

t(0)=A(0)/1=0

t(n)=t(n-1)+$\frac{2}{n+1}$

t(n)=t(n-2)+$\frac{2}{n}+\frac{2}{n+1}$

t(n)=t(n-3)+$\frac{2}{n-1}+\frac{2}{n}+\frac{2}{n+1}$

. . . .

t(n)=$\sum_{i=2}^{n}\frac{2}{i+1}$ → Harmonic series

A(n)=Θ(nlogn)

4) Finding Median Worst Case:

An extremely unbalanced partition (with 1 part empty and the other part (n-1) elements)

C_{worst}(n)=(n-1)+(n-2)+…+1=n.(n-1)/2 = Θ(n^2)